

MODELO DE COSTOS PARA LA FIJACIÓN DE CARGO DE INTERCONEXIÓN TOPE POR TERMINACIÓN DE LLAMADAS EN LA RED DE TELEFONÍA FIJA LOCAL, EN LAS MODALIDADES DE TIEMPO DE OCUPACIÓN Y CAPACIDAD

Elaborado por:

Gerencia de Políticas Regulatorias
Organismo Supervisor de la Inversión Privada en Telecomunicaciones

Dirección:

Lima, Perú
Oficina Principal: Calle de La Prosa 136 Lima 41 - Telf. 2251313 - Fax 4751816
Página Web: <http://www.osiptel.gob.pe/>

CONFIGURACIONES PREVIAS

Configuración del Entorno de Trabajo

Esta sección realiza configuraciones del entorno de trabajo para el Kernel del Software Mathematica

■ Carga el paquete Geodesy

Este paquete es utilizado para el cálculo de distancias entre dos puntos sobre la superficie de la Tierra. Dichos puntos corresponden a la ubicación de las centrales de conmutación

```
<< Geodesy`
```

■ Desactiva opciones de Sintaxis

El Software Mathematica es capaz de identificar nombres similares de variables (Símbolos) y emitir advertencias al respecto cuandoel Kernel realiza la ejecución del código.
La existencia de nombres similares es aceptada para la programación del presente modelo

```
Off[General::spell1];  
Off[General::spell];
```

■ Establece el directorio de trabajo

El modelo utiliza diversos archivos de texto como entrada. En esta instrucción se establece el directorio donde se encuentran ubicados dichos archivos

```
SetDirectory[  
  "D:\\Data\\Confidencial\\Fija2006\\Modelo de Costos\\Modelo Integral\\Aceptaciones\\AceptacionesTdP";
```

CARGA DE PARÁMETROS GENERALES

■ Identificación de Areas Locales: Departamentos

Listado de los departamentos y les asigna un número correlativo

AYACUCHO = 1; HUANCAVELICA = 2; HUANUCO = 3; ICA = 4; JUNIN = 5; LIMA = 6; LORETO = 7; PASCO = 8; SANMARTIN = 9; UCAYALI = 10; AMAZONAS = 11; ANCASH = 12; CAJAMARCA = 13; LALIBERTAD = 14; LAMBAYEQUE = 15; PIURA = 16; TUMBES = 17; APURIMAC = 18; AREQUIPA = 19; CUSCO = 20; MADREDEDIOS = 21; MOQUEGUA = 22; PUNO = 23; TACNA = 24;
--

Parámetros de Centrales de Conmutación

■ Codificación de Centrales Tandems Departamentales

Identifica las centrales Tandem dentro del listado de 585

CHACHAPOYASX=1; HUARAZX=5; ABANCAYX=36; AREQUIPACENTROX=39; AYACUCHOX=86; CAJAMARCAX=92; CUSCOCENTROX=110; HUANCAVELICAX=124; HUANUCOX=128; ICACENTROX=132; HUANCAYOCENTROX=150; TRUJILLOIX=183; CHICLAYOCENTROX=223; TANDEMLIMAX= 248; IQUITOSCENTROX=481; PTOMALDONADOX=491; ILOCENTROX=492; CERRODEPASCOX=498; PIURACENTROX=504; JULIACAX=534; TARAPOTOX=549; TACNACENTROX=561; TUMBESX=574; PUCALLPACENTROX=581;
CentralRef=TANDEMLIMAX;

■ Codificación de Centrales Cabeceras (Provincias)

Listado de las cabeceras y les asigna un código correlativo

ABANCAYP=1; AREQUIPACAYMAP=2; AREQUIPACENTROP=3; AREQUIPAPRXP=4; AYACUCHOP=5; BARRANCONEAXP=6; CAJAMARCAP=7; CALLAOP=8; CUSCOCENTROP=9; CHACHAPOYASP=10; CHICLAYOCENTROP=11; CHIMBOTECENTROP=12; ELCERCADO1P=13; ELCERCADO2P=14; HIGUERETA1P=15; HIGUERETA2P=16; HIGUERETANEAXP=17; HUACHOP=18;
--

HUANCVELICAP=19; HUANCAYOCENTROP=20; HUANCAYOPRXP=21; HUANUCOP=22; HUARAZP=23; ICACENTROP=24; ICAPRXP=25; ILOCENTROP=26; IQUITOSCENTROP=27; JAENP=28; JULIACAP=29; LAVICTORIANEAXP=30; LINCEP=31; LINCENEAXP=32; LOSOLIVOS1P=33; LOSOLIVOS2P=34; LOSOLIVOS3P=35; LOSOLIVOSNEAXP=36; MAGDALENAP=37; MAGDALENANEAXP=38; MIRAFLORES1P=39; MIRAFLORES2P=40; MIRAFLORESNEAXP=41; MONTERRICO1P=42; MONTERRICO2P=43; MONTERRICO3P=44; MONTERRICONEAXP=45; PIURACENTROP=46; PIURAPRXP=47; PTOMALDONADOP=48; PUCALLPACENTROP=49; PUNOPRXP=50; QUILLABAMBAP=51; SANBORJANEAXP=52; SANISIDRO1P=53; SANISIDRO2P=54; SANISIDRONEAXP=55; SANJOSE1P=56; SANJOSE2P=57; SANJOSE3P=58; SANJOSENEAXP=59; SANJUANNEAXP=60; TACNACENTROP=61; TACNAPRXP=62; TANDEMLIMAP=63; TARAPOTOP=64; TARMAP=65; TRUJILLOIP=66; TRUJILLOIIP=67; TRUJILLOPRXP=68; TUMBESP=69; WASHINGTON1P=70; WASHINGTON2P=71; WASHINGTONNEAXP=72; ZARATE1P=73; ZARATE2P=74; CERRODEPASCOP=75; ATALAYAP=76; CABALLOCOCHAP=77; CONTAMANAP=78; LAGUNASP=79; NAUTAP=80; REQUENAP=81; AGUASCALIENTESP=82; BELLAVISTAP=83; CABANAP=84; CAJATAMBOP=85; CORACORAP=86; IQUITOSCENTROP=87; JUANJUIP=88; PUQUIOP=89; SANFRANCISCOP=90; SAPOSOAP=91; TINGOMARIAP=92; TOCACHENUEVOP=93; UCHIZAP=94; ZARATE3P=95; LOSOLIVOS4P=96; LOSOLIVOS5P=97; SANISIDRO3P=98; MIRAFLORES3P=99; HIGUERETAP=100;
--

■ Identificación de Centrales Tandems (Departamentales) por Nombre

```
deptTXTd = { "AYACUCHO", "HUANCAVELICA", "HUANUCO", "ICA",  
             "JUNIN", "LIMA", "LORETO", "PASCO", "SANMARTIN", "UCAYALI", "AMAZONAS", "ANCASH",  
             "CAJAMARCA", "LALIBERTAD", "LAMBAYEQUE", "PIURA", "TUMBES", "APURIMAC",  
             "AREQUIPA", "CUSCO", "MADREDEDIOS", "MOQUEGUA", "PUNO", "TACNA"};
```

■ Identificación de Centrales Cabeceras (Provincias) por Nombre

```
provTXTp = { "ABANCAYP", "AREQUIPACAYMAP", "AREQUIPACENTROP", "AREQUIPAPRXP", "AYACUCHOP", "BARRANCONEEXP",  
             "CAJAMARCAP", "CALLAOP", "CUSCOCENTROP", "CHACHAPOYASP", "CHICLAYOCENTROP", "CHIMBOTECENTROP",  
             "ELCERCADO1P", "ELCERCADO2P", "HIGUERETA1P", "HIGUERETA2P", "HIGUERETANEEXP", "HUACHOP", "HUANCAVELICAP",  
             "HUANCAYOCENTROP", "HUANCAYOPRXP", "HUANUCOP", "HUARAZP", "ICACENTROP", "ICAPRXP", "ILOCENTROP",  
             "IQUITOSCENTROP", "JAENP", "JULIACAP", "LAVICTORIANEEXP", "LINCEP", "LINCENEEXP", "LOSOLIVOS1P",  
             "LOSOLIVOS2P", "LOSOLIVOS3P", "LOSOLIVOSNEEXP", "MAGDALENAP", "MAGDALENANEEXP", "MIRAFLORES1P",  
             "MIRAFLORES2P", "MIRAFLORESNEEXP", "MONTERRICO1P", "MONTERRICO2P", "MONTERRICO3P", "MONTERRICONEEXP",  
             "PIURACENTROP", "PIURAPRXP", "PTOMALDONADOP", "PUCALLPACENTROP", "PUNOPRXP", "QUILLABAMBAP",  
             "SANBORJANEEXP", "SANISIDRO1P", "SANISIDRO2P", "SANISIDRONEEXP", "SANJOSE1P", "SANJOSE2P", "SANJOSE3P",  
             "SANJOSENEEXP", "SANJUANNEEXP", "TACNACENTROP", "TACNAPRXP", "TANDEMLIMAP", "TARAPOTOP", "TARMAP",  
             "TRUJILLOIP", "TRUJILLOIIP", "TRUJILLOPRXP", "TUMBESP", "WASHINGTON1P", "WASHINGTON2P", "WASHINGTONNEEXP",  
             "ZARATE1P", "ZARATE2P", "CERRODEPASCOP", "ATALAYAP", "CABALLOCOCHAP", "CONTAMANAP", "LAGUNASP",  
             "NAUTAP", "REQUENAP", "AGUASCALIENTESP", "BELLAVISTAP", "CABANAP", "CAJATAMBOP", "CORACORAP",  
             "IQUITOSCENTROP", "JUANJUIP", "PUQUIOP", "SANFRANCISCOP", "SAPOSOAP", "TINGOMARIAP", "TOCACHENUEVOP",  
             "UCHIZAP", "ZARATE3P", "LOSOLIVOS4P", "LOSOLIVOS5P", "SANISIDRO3P", "MIRAFLORES3P", "HIGUERETAP"};
```

■ Identificación y Codificación de Función de Central

Esquema de identificación lógico entre Centrales Cabeceras y URAS (Unidades Remotas, tanto Analógicas como Digitales)

```
CABECERA=1;  
URD=0;
```

■ Identificación y Codificación de Tecnologías de transmisión asociadas a cada Central de Conmutación

■ Todas las Tecnologías de Transmisión utilizadas

Listado de Tecnologías identificadas por un número

```
Satelite=1;  
Enterrado=2;  
Radio=3;  
NA=4;  
Urbana=5;  
Aereo=6;
```

■ Identificación especial de Tecnología de Fibra Óptica

Donde diga Fiberoptic esta haciendo referencia a la tecnología de fibra enterrada

```
Fiberoptic=2;
```

Lectura de Archivo de Texto "input_TdP.txt"

■ Carga datos de Archivo de Texto "input_TdP.txt"

■ Abre Archivo de Texto "input_TdP.txt"

```
OpenRead ["input_TdP.txt"];
```

■ Lee archivo y lo carga en variable auxiliar "d"

```
d = N[ReadList ["input_TdP.txt", Number]];
```

El archivo de texto ha sido leído y la información de las variables que contiene ya se encuentra en memoria para ser utilizada. En las secciones siguientes se asignan estos valores a las variables respectivas

Parámetros Generales

```
locb = d[[1]] ;
calculaservicio = d[[2]] ;
(* Si vale 2 se calcula coste servicio circuitos de ITX (sólo intradepartamental) *)
(* Si es 3,4 ó 5 se calcula coste de circuitos arrendados LD rango A, B o C respectivamente *)
calculaservicioLD = calculaservicio;
(* crea variable calculaservicioLD y le asigna el valor de la variable calculaservicio *)
accesodistporE1 = d[[3]] ;
(* Distancia media por E1 entre el local del operador y la central de TDP *)
prob = d[[4]] ;
(* Probabilidad de Bloqueo: Calidad *)
numMHzporMb = d[[52]] ;
(* Número de MhZ de ancho de banda al codificar 1 Mbps de datos para Tx Satelital *)
factorNoLinealUrban =  $\frac{1}{d[[57]]}$  ;
(* Factor de no linealidad de fibra óptica canalizada *)
factorNoLinealNoUrban =  $\frac{1}{d[[58]]}$  ;
(* Factor de no linealidad de fibra óptica enterrada *)
factorNoLinealRadio =  $\frac{1}{d[[59]]}$  ;
(* Factor de no linealidad de los enlaces de radio *)
factnolinLD = d[[69]] ;
(* Factor de no linealidad de los enlaces de FO entre los PDIs *)
edifsupportdeprfrac = d[[60]] ;
(* Depreciación de los edificios *)
fibertransdeprfrac = d[[61]] ;
(* Depreciación de la fibra óptica *)
petransdeprfrac = d[[62]] ;
(* Depreciación de la planta externa *)
trafAdicNoTarif = d[[63]] ;
(* Plus porcentual por el tráfico no tarificado *)
compFOLD = d[[68]] ;
(* Factor de compartición de los enlaces de fibra entre los PDIs con la red intradepartamental*)
compfiburb = d[[54]] ;
(* Factor de compartición de fibra canalizada con la red de acceso *)
factalq = d[[50]] ;
(* Factor de alquiler intradepartamental *)
factalqLD = d[[5]] ;
(* Factor de Alquiler Internodal o entre los PDI'
s: Qué % de los circuitos interdepartamentales son alquilados *)
```

Parámetros de Señalización

```
signalfrac=d[[24]] ;
(* Inversión en Señalización como % de la Inversión en Conmutación *)
```

Parámetros de Transmisión

■ Parámetros de Transmisión Generales

```
hop=d[[7]];
(* Distancia en Km entre las repetidoras de radio *)
hopfibra=d[[67]];
(* Distancia entre las repetidoras de fibra óptica *)
urbanfill=d[[6]];
(* Grado de Utilización de la F.O. urbana o canalizada *)
radiofill=d[[8]];
(* Grado de Utilización de los enlaces de Radio *)
fiberfill=d[[13]];
(* Grado de utilización de la Fibra Óptica Enterrada *)
ConsideraPRIM=d[[9]];
(* Para un valor de 0 se emplean conexiones en estrella, para un valor de 1 se emplean conexiones en cadena y para un valor de 3 se emplean conexione en cadena con protección *)
consideraRD = d[[49]];
(* Si vale 1 si se consideran las rutas directas entre las centrales cabeceras de Lima, si vale 0 no se consideran *)
interdinvpermin=d[[11]];
(* Inversión adicional por MDU interdepartamental por las llamadas satelitales que pasan por Lima *)
compresDCME=d[[12]];
(* Compresión de equipo DCME satélite: menor o igual a 1*)
redundancyfactor= d[[48]];
(* Factor de redundancia de la ruta a las centrales Tandem de Lima *)
proteccionLD=d[[70]];
(* Factor de protección de F.O. entre los PDIs: Si vale 1 se considera que no existe protección de F.O. y que la protección es por radio *)
```

■ Parámetros de Transmisión de Fibra

```
TxFibraCosteCableKm=d[[26]];
(* Precio por Km de cable incluyendo planta externa, Fibra Enterrada*)
TxFibraCosteCableAereoKm=2676.49641458801;
(* Precio por Km de cable, Fibra Aérea*)
costedelafibraurbKm=d[[71]];
(* Precio por Km de la Fibra canalizada (sólo cable) *)
TxFibraCosteEmpalme= d[[27]];
(* Precio por empalme de F.O. *)
TxFibraDistEntreEmpalmesKm=d[[28]];
(* Distancia en Km entre los empalmes de F.O. *)
TxFibraCosteCanalizacionKm=d[[29]];
(* Precio por Km de canalización *)
TxFibraCosteCamara=d[[30]];
(* Precio de una cámara de F.O. *)
TxFibraDistEntreCamarasKm=d[[31]];
(* Distancia en Km entre cámaras *)
TxFibraCosteTrituboKm=d[[32]];
(* Precio por Km del tritubo: es cero porque ya esta incluido en TxFibraCosteCableKm *)
TxFibraCosteCableKmLD=d[[72]];
(* Precio por Km de cable incluyendo planta externa para la fibra enterrada entre los PdIs *)
TxFibraLimites=63{1,4,16,64};
(* Niveles en E1s *)
TxFibraCosteFijo={d[[33],d[[34],d[[35],d[[55]]];
(* Vector de precio de los equipos de Tx según nivel de capacidad *)
TxFibraCosteTributario={d[[36],d[[37],d[[38],d[[56]]];
(* Vector de precios de los tributarios según nivel de capacidad *)
```

■ Parámetros de Transmisión de Radio

```
TxRadioLimites={4,16,32,63};
(* Define los límites de cada uno de los niveles de capacidad entre los cuales se puede optar:
4x2(1+0);16x2(1+1);16x2(2+1)y STM1 (63 E1s) *)
TxRadioCosteFijo={d[[39],d[[40],d[[41],d[[42]]];
(* Vector de precios de los equipos de Tx de Radio en PDH para los siguientes niveles de
capacidad: 4x2(1+0), 16x2(1+1), 16x2(2+1), STM-1(Radio+ADM) *)
TxRadioCosteTributario={0,0,0,d[[43]]};
(* Vector de precios de los tributarios en radio según el nivel de capacidad *)
TxRadioCosteRep={d[[44],d[[45],d[[46],d[[47]]];
(* Vector de precios de las repetidoras de radio *)
```


Parámetros de Costos

```
edifsupportinvfrac=d[[10]];
(* Inversión en edificios como un % de la inversión en red: Este factor no debe existir pues ya existe un factor
por Inversion en Soporte en General *)
supportinvfrac=d[[14]];
(* Inversión en soporte como % de la inversión en red *)
switchdeprfrac= d[[15]];
(* Depreciación de los equipos de conmutación *)
transdeprfrac=d[[16]];
(* Depreciación de los equipos de transmisión *)
supportdeprfrac=d[[17]];
(* Depreciación de la planta de soporte *)
switchmaintfrac=d[[18]];
(* Gastos de mantenimiento en conmutación como % de la Inversión en conmutación *)
transmaintfrac=d[[19]];
(* Gastos de mantenimiento en transmisión como % de la Inversión en transmisión *)
supportmaintfrac=d[[20]];
(* Gastos de mantenimiento de la planta de soporte como % de la inversión en soporte *)
overheadfrac=d[[21]];
(* Costo de Overhead como % de la inversión en red *)
networkopfrac=d[[22]];
(* Costo de operación de red como % de la inversión en red *)
retcap=d[[23]];
(* Costo de oportunidad del capital: WACC *)
satantena=d[[25]];
(* Precio de Antena Satelital *)
sateqtx= d[[51]];
(* Precio de equipo de transmisión satelital *)
costesatporMHz= d[[53]];
(* Costo anual de alquiler de 1 MHZ de ancho de banda satelital *)
AñadidoCxPorKm=d[[64]];
(* Costo por Km del transporte o desplazamiento de los equipos de Cx a provincias *)
AñadidoTrxPorKm=d[[65]];
(* Costo por Km del transporte o desplazamiento de los equipos de Tx a provincias *)
TxFibraCosteTerminacion=d[[66]];
(* Costo de las terminaciones de fibra óptica *)
factrednd=d[[73]];
(* Factor de Redundancia para los enlaces de ADSL - Ojo, que podría no ser un factor único *)
locchannelinvpertrunk=d[[74]];
(* Factor Rescatado para Conmutación (era el Nro 5) *)
```

Lectura de Archivo de Texto "input_centralesfinal.txt"

En el archivo de texto "input_centralesfinal.txt" se encuentra el detalle de cada central de conmutación utilizada en el modelo. Dichos parámetros deben ser leídos y cargados de forma adecuada para su posterior uso. Se ha considerado conveniente leerlos en una variable de tipo matriz y posteriormente asignarlo a cada variable específica.

■ Desactiva Verificación de Sintaxis

El software Mathematica es capaz de detectar errores de sintaxis cuando se utiliza la Función ToExpression. Esta opción se ha deshabilitado para permitir la asignación del archivo leído a la variable auxiliar utilizada

```
Off[ToExpression::"sntxi"];
```

■ Carga datos de Archivo de Texto "input_centralesfinal.txt"

■ Abre Archivo de Texto "input_centralesfinal.txt"

```
OpenRead["input_centralesfinal.txt"];
```

■ Lee archivo y lo carga en variable auxiliar "y"

```
y = ReadList["input_centralesfinal.txt", Word, RecordLists -> True];
```

■ El archivo leído es convertido a formato de expresión para su posterior uso en variable auxiliar "z"

```
z = ToExpression[y];
(* Define la matriz "z" que contienen los datos de las centrales ya leídos *)
```

El archivo de texto ha sido leído y la información de las variables que contiene ya se encuentra en memoria para ser utilizada. En las secciones siguientes se asignan estos valores a las variables respectivas

Detalle de Centrales de Conmutación

■ Obtiene el número de centrales de conmutación (variable auxiliar "nx")

```
nx=Length[z];
(* Define el número de datos que contiene la matriz z *)
```

■ Detalle de Centrales de Conmutación

```
idx = Table[z[[ix]][[1]], {ix, nx}];
(* Especifica para cada central su ID *)
idinvertx = Table[0, {ix, nx}];
Do[idinvertx[[idx[[ix]]] = ix, {ix, nx}];
Traducir[Vector_] := Module[{Vector1, n, dim}, n = Dimensions[Vector][[1]];
  dim = Dimensions[Vector][[2]];
  Vector1 = Table[Table[0, {d, dim}], {i, n}];
  Do[If[dim == 2, Vector1[[i, 1]] = idinvertx[[Vector[[i, 1]]];
    Vector1[[i, 2]] = Vector[[i, 2]];];
  If[dim == 3, Vector1[[i, 1]] = idinvertx[[Vector[[i, 1]]];
    Vector1[[i, 2]] = idinvertx[[Vector[[i, 2]]];
    Vector1[[i, 3]] = Vector[[i, 3]];];, {i, n}];
  Vector1];
(* Especifica para cada ID la posición de la central *)
deptx = Table[z[[ix]][[3]], {ix, nx}];
(* Especifica para cada central el departamento al cual pertenece *)
provx = Table[z[[ix]][[4]], {ix, nx}];
(* Especifica para cada central la provincia a la cual pertenece *)
longdegx = Table[z[[ix]][[5]], {ix, nx}];
(* Especifica para cada central la coordenada: longitud en grados *)
longminx = Table[z[[ix]][[7]], {ix, nx}];
(* Especifica para cada central la coordenada: longitud en minutos *)
longsecx = Table[z[[ix]][[9]], {ix, nx}];
(* Especifica para cada central la coordenada: longitud en segundos *)
latdegx = Table[z[[ix]][[6]], {ix, nx}];
(* Especifica para cada central la coordenada: latitud en grados *)
latminx = Table[z[[ix]][[8]], {ix, nx}];
(* Especifica para cada central la coordenada: latitud en minutos *)
latsecx = Table[z[[ix]][[10]], {ix, nx}];
(* Especifica para cada central la coordenada: latitud en segundos *)
linesx = Table[z[[ix]][[11]], {ix, nx}];
(* Especifica para cada central el número de líneas *)
systemx = Table[z[[ix]][[12]], {ix, nx}];
(* Especifica para cada central su función: si es 1 la central es cabecera, si es 0 la central es una URA *)
transtechx = Table[z[[ix]][[2]], {ix, nx}];
(* Especifica para cada central la tecnología de transmisión *)
esAnillopx = Table[z[[ix]][[14]], {ix, nx}];
Do[If[esAnillopx[[ix]] == 1 & transtechx[[ix]] != Urbana, transtechx[[ix]] = Urbana];, {ix, nx}];
(* Especifica para cada central si forma parte del anillo provincial *)
esAnillodx = Table[z[[ix]][[15]], {ix, nx}];
(* Especifica para cada central si forma parte del anillo departamental *)
```

Imprime Parámetros

■ Imprime Parámetros Generales

```
Print[TableForm[{locb, calculaservicio, calculaservicio, accesodistporE1, prob, numMHZporMb, factorNoLinealUrban, factorNoLinealNoUrban, factorNoLinealRadio, factnolinLD, edifsupportdeprfrac, fibertransdeprfrac, petransdeprfrac, trafAdicNoTarif, compFOLD, compfiburb, factalq, factalqLD}, TableHeadings→
{{"locb", "calculaservicio", "calculaservicioLD", "accesodistporE1", "prob", "numMHZporMb", "factorNoLinealUrban", "factorNoLinealNoUrban", "factorNoLinealRadio", "factnolinLD", "edifsupportdeprfrac", "fibertransdeprfrac", "petransdeprfrac", "trafAdicNoTarif", "compFOLD", "compfiburb", "factalq", "factalqLD"}}]];

```

■ Imprime Parámetros de Señalización

```
Print[TableForm[{signalfrac, ""}, TableHeadings→{{"signalfrac", ""}}]];

```


Imprime Parámetros de Transmisión

■ Imprime Parámetros de Transmisión Generales

```
Print[TableForm[{hop,hopfibra,urbanfill,radiofill,fiberfill,ConsideraPRIM,consideraRD,interdinvpermin,compresDCME,
redundancyfactor,proteccionLD},TableHeadings→
{{"hop","hopfibra","urbanfill","radiofill","fiberfill","ConsideraPRIM","consideraRD","interdinvpermin","compresDCM
E","redundancyfactor","proteccionLD"}}]]];
```

■ Imprime Parámetros de Transmisión de Fibra

```
Print[TableForm[{TxFibraCosteCableKm,TxFibraCosteCableAereoKm,costedelafibraurbKm,TxFibraCosteEmpalme,TxFibraDistE
ntreEmpalmesKm,TxFibraCosteCanalizacionKm,TxFibraCosteCamara,TxFibraDistEntreCamarasKm,TxFibraCosteTrituboKm,TxFib
raCosteCableKmLD,TxFibraLimites,TxFibraCosteFijo,TxFibraCosteTributario},TableHeadings→
{{"TxFibraCosteCableKm","TxFibraCosteCableAereoKm","costedelafibraurbKm","TxFibraCosteEmpalme","TxFibraDistEntreEm
palmesKm","TxFibraCosteCanalizacionKm","TxFibraCosteCamara","TxFibraDistEntreCamarasKm","TxFibraCosteTrituboKm","T
xFibraCosteCableKmLD","Capacidades en SDH","TxFibraCosteFijo","TxFibraCosteTributario"}}]]];
```

■ Imprime Parámetros de Transmisión de Radio

```
Print[TableForm[{TxRadioLimites,TxRadioCosteFijo,TxRadioCosteTributario,TxRadioCosteRep},TableHeadings→
{{"Capacidades para Radio","TxRadioCosteFijo","TxRadioCosteTributario","TxRadioCosteRep"}}]]];
```

■ Imprime Parámetros de Costos

```
Print[TableForm[{edifsupportinvfrac,supportinvfrac,switchdeprfrac,transdeprfrac,supportdeprfrac,switchmaintfrac,tr
ansmaintfrac,supportmaintfrac,overheadfrac,networkopfrac,retcap,satantena,sateqtx,costesatporMHz,AñadidoCxPorKm,Añ
adidoTrxPorKm,TxFibraCosteTerminacion,factrednd,locchannelinvpertrunk},TableHeadings→
{{"edifsupportinvfrac","supportinvfrac","switchdeprfrac","transdeprfrac","supportdeprfrac","switchmaintfrac","tran
smaintfrac","supportmaintfrac","overheadfrac","networkopfrac","retcap","satantena","sateqtx","costesatporMHz","Aña
didoCxPorKm","AñadidoTrxPorKm","TxFibraCosteTerminacion","factrednd","locchannelinv"}}]]];
```

■ Imprime Parámetros de Detalle de Centrales de Conmutación

■ Imprime Número de Centrales de Conmutación

```
Print["nx"];
Print[nx];
```

■ Imprime Detalle de Centrales de Conmutación

```
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],deptx[[ix]],provx[[ix]],longdegx[[ix]],longminx[[ix]],longsecx[[ix]],latdegx[[
ix]],latminx[[ix]],latsecx[[ix]],linesx[[ix]],systemx[[ix]],transtechx[[ix]],esAnillopx[[ix]],esAnillodx[[
ix]],{ix,nx}},TableAlignments→Right,TableSpacing→{0,1},TableDirections→{Column,Row},TableHeadings→
{None,{"ix","idx","Central","deptx","provx","longdegx","longminx","longsecx","latdegx","latminx","latsecx","linesx
","systemx","transtechx","esAnillopx","esAnillodx"}}]]]
```

Creación de Variables Auxiliares

■ Número de datos a nivel de departamento (variable auxiliar "nd")

```
nd=24;
(* Número de datos a nivel de cálculos y resultados departamentales *)
```

■ Número de datos a nivel de provincia (variable auxiliar "np")

```
np=Max[provx];
(* Número de datos a nivel de cálculos y resultados provinciales *)
```

■ Conjunto de Vectores Cero (variable auxiliar "zerox", "zerop", "zerod")

- Vector Cero de Centrales (variable auxiliar "zerox")

```
zerox=Table[0,{ix,nx}]; (* vector de 585 ceros: Número de centrales *)
```

- Vector Cero de Provincias (variable auxiliar "zerop")

```
zerop=Table[0,{ip,np}]; (* vector de 94 ceros: Número de provincias *)
```

- Vector Cero de Departamentos (variable auxiliar "zerod")

```
zerod=Table[0,{id,nd}]; (* vector de 24 ceros: Número de departamentos *)
```

■ Conjunto de Vectores Uno (variable auxiliar "onex", "onep ", "oned")

- Vector Uno de Centrales (variable auxiliar "onex")

```
onex=Table[1,{ix,nx}]; (* vector de 585 unos: Número de centrales *)
```

- Vector Uno de Provincias (variable auxiliar "onep")

```
onep=Table[1,{ip,np}]; (* vector de 94 unos: Número de provincias *)
```

- Vector Uno de Departamentos (variable auxiliar "oned")

```
oned=Table[1,{id,nd}]; (* vector de 24 unos: Número de departamentos *)
```

■ Vector de Nodos que pertenecen a Lima (variable auxiliar "lima1d")

```
lima1d=Table[If[id==LIMA,1,0],{id,nd}];  
(* Identifica cada nodo que pertenece a Lima como "1" y los demás como "0" *)
```

■ Vector de Departamentos asociados a cada Provincia (variable auxiliar "deftp")

```
deftp=zerop;  
Do[deftp[[provx[ix]]]=deptx[[ix]},{ix,nx}]  
(* Especifica para cada provincia a que departamento pertenece *)
```

■ Conjunto de vectores Lima/Provincia (variable auxiliar "limax", "lima1x", "ropx", "rop1x")

- Vector de Nodos que pertenecen a Lima - True/False (variable auxiliar "limax")

```
limax=Table[deptx[[ix]]==LIMA,{ix,nx}];  
(* Identifica cada nodo que pertenece a Lima como "True" y los demás como "False" *)
```

- Vector de Nodos que pertenecen a Lima - 1/0 (variable auxiliar "lima1x")

```
lima1x=Table[If[limax[[ix]],1,0],{ix,nx}];  
(* Identifica cada nodo que pertenece a Lima como "1" y los demás como "0" *)
```

- Vector de Nodos que pertenecen a Provincias - True/False (variable auxiliar "ropx")

```
ropx=Table[¬limax[[ix]},{ix,nx}];  
(* Identifica cada nodo que pertenece a provincias como "True" y los de Lima como "False" *)
```

- Vector de Nodos que pertenecen a Provincias - 1/0 (variable auxiliar "rop1x")

```
rop1x=Table[If[ropx[[ix]],1,0],{ix,nx}];
(* Identifica cada nodo que pertenece a provincias como "1" y los de Lima como "0" *)
```

Imprime Variables Auxiliares

- Imprime Número de datos a nivel de departamento

```
Print["nd"];
Print[nd];
```

- Imprime Número de datos a nivel de provincia

```
Print["np"];
Print[np];
```

- Imprime Vector de Nodos que pertenecen a Lima

```
Print["lima1d"];
Print[TableForm[Table[{id,deptTXtd[[id]],lima1d[[id]]},{id,nd}],TableDirections→{Column,Row},TableSpacing→{0,1},TableHeadings→{None,{"id","Departamento","lima1d"}}]]];
```

- Imprime Vector de Departamentos asociados a cada Provincia

```
Print["deptp"];
Print[TableForm[Table[{ip,provTXtp[[ip]],deptp[[ip]],deptTXtd[[deptp[[ip]]]},{ip,np}],TableDirections→{Column,Row},TableSpacing→{0,1},TableHeadings→{None,{"ip","Provincia","id","Departamento"}}]]];
```

- Imprime Conjunto de vectores Lima/Provincia

```
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],limax[[ix]],lima1x[[ix]],ropx[[ix]],rop1x[[ix]]},{ix,nx}],TableSpacing→{0,1},TableDirections→{Column,Row},TableHeadings→{None,{"ix","idx","Nombre Central","limax","lima1x","ropx","rop1x"}}]]];
```

Definición de Funcion auxiliar

- Define función auxiliar "distance"

Presenta y establece como se estimara la distancia entre dos puntos utilizando la función denominada [SphericalDistance](#)

```
distance[ix1_, ix2_] := Module[{vect1, vect2, dist},
  vect1 = {latdegx[[ix1]] +  $\frac{\text{latminx}[[ix1]]}{60} + \frac{\text{latsecx}[[ix1]]}{3600}$ , longdegx[[ix1]] +  $\frac{\text{longminx}[[ix1]]}{60} + \frac{\text{longsecx}[[ix1]]}{3600}$ };
  vect2 = {latdegx[[ix2]] +  $\frac{\text{latminx}[[ix2]]}{60} + \frac{\text{latsecx}[[ix2]]}{3600}$ , longdegx[[ix2]] +  $\frac{\text{longminx}[[ix2]]}{60} + \frac{\text{longsecx}[[ix2]]}{3600}$ };
  If[vect1 == vect2, dist = 0, dist = SphericalDistance[vect1, vect2]]; dist];
```

ESTIMACIÓN DE DISTANCIAS

Calculo de la distancia entre nodos (centrales)

■ Identificación de Nodos por cada tecnología de transmisión

■ Identificación de Nodos que utilizan tecnología satelital (variables satx, sat1x)

```
satx=Table[False,{ix,nx}];
satx[[1]]=transtechx[[1]]==Satelite;
Do[satx[[ix]]=If[provx[[ix]]==provx[[ix-1]],satx[[ix-1]],(*deptx[[ix]]#LIMA ^*) deptx[[ix]]#MADREDEDIOS ^ transtechx[[ix]]==Satelite)];,{ix,2,nx}];
(* Identifica los nodos que usan tecnología de Tx satelital con "True" y los demás con "False" , ojo que en Lima no pueden haber nodos que usen tecnología satelital *)
sat1x=Table[If[satx[[ix]],1,0],{ix,nx}];
(* Identifica los nodos que usan tecnología de Tx satelital con "1" y los demás con "0", ojo que en Lima no pueden haber nodos que usen tecnología satelital *)
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],satx[[ix]],sat1x[[ix]]},{ix,nx}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre Central","satx","sat1x"}}]]];
```

■ Identificación de Nodos que utilizan tecnología de fibra optica (variables fiberx, fibert1x)

```
fiberx=Table[False,{ix,nx}];
Do[fiberx[[ix]]=transtechx[[ix]]==Enterrado v transtechx[[ix]]==Urbana v transtechx[[ix]]==Aereo,{ix,nx}];
(* Identifica los nodos que usan fibra como tecnología de Tx con "True" y los demás con "False" *)
fiber1x=Table[If[fiberx[[ix]],1,0],{ix,nx}];
(* Identifica los nodos que usan fibra como tecnología de Tx con "1" y los demás con "0"*)
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],fiberx[[ix]],fiber1x[[ix]]},{ix,nx}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre Central","fiberx","fiber1x"}}]]];
```

■ Identificación de Centrales Tandem por Area Local (Departamento)

■ Identificación de Centrales Tandem por Cada Area Local (Departamento) (variable tandemd)

```
Print["tandemd"];
tandemd=zerod;
tandemd[[AMAZONAS]]=idinvertx[[CHACHAPOYASX]];
tandemd[[ANCASH]]=idinvertx[[HUARAZX]];
tandemd[[APURIMAC]]=idinvertx[[ABANCAYX]];
tandemd[[AREQUIPA]]=idinvertx[[AREQUIPACENTROX]];
tandemd[[AYACUCHO]]=idinvertx[[AYACUCHOX]];
tandemd[[CAJAMARCA]]=idinvertx[[CAJAMARCAX]];
tandemd[[CUSCO]]=idinvertx[[CUSCOCENTROX]];
tandemd[[HUANCAVELICA]]=idinvertx[[HUANCAVELICAX]];
tandemd[[HUANUCO]]=idinvertx[[HUANUCOX]];
tandemd[[ICA]]=idinvertx[[ICACENTROX]];
tandemd[[JUNIN]]=idinvertx[[HUANCAYOCENTROX]];
tandemd[[LALIBERTAD]]=idinvertx[[TRUJILLOIX]];
tandemd[[LAMBAYEQUE]]=idinvertx[[CHICLAYOCENTROX]];
tandemd[[LIMA]]=idinvertx[[TANDEMLIMAX]];
tandemd[[LORETO]]=idinvertx[[IQUITOSCENTROX]];
tandemd[[MADREDEDIOS]]=idinvertx[[PTOMALDONADOX]];
tandemd[[MOQUEGUA]]=idinvertx[[ILOCENTROX]];
tandemd[[PASCO]]=idinvertx[[CERRODEPASCOX]];
tandemd[[PIURA]]=idinvertx[[PIURACENTROX]];
tandemd[[PUNO]]=idinvertx[[JULIACAX]];
tandemd[[SANMARTIN]]=idinvertx[[TARAPOTOX]];
tandemd[[TACNA]]=idinvertx[[TACNACENTROX]];
tandemd[[TUMBES]]=idinvertx[[TUMBESX]];
tandemd[[UCAYALI]]=idinvertx[[PUCALLPACENTROX]];
Print[TableForm[Table[{id,deptXTd[id],tandemd[id],y[[tandemd[id]],13]},{id,nd}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"id","Nombre Departamento","id Tandem","Nombre Tandem"}}]]];
(* Identifica para cada departamento cuál es la central que será considerada Tandem *)
```

■ Identificación de Central Tandem por cada Central (variables tandemx, tandem1x)

```
tandemx=Table[tandemd[[deptx[[ix]]]],{ix,nx}];
(* Para cada central se especifica cuál es la Tandem de su departamento *)
tandem1x=Table[If[tandemx[[ix]]==ix,1,0],{ix,nx}];
(* Para aquellas centrales que son Tandem las identifica con "1" y las que no son con "0" *)
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],tandemx[[ix]],idx[[tandemx[[ix]]]],y[[tandemx[[ix]],13]],tandem1x[[ix]]},{ix,nx}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre Central","tandemx","idtandemx","Nombre Tandem","tandem1x"}}]]];
```

Identificación de Centrales Cabeceras (variables host1x, hostx)

```
host1x=Table[Systemx[[ix]],{ix,nx}];
(* Para aquellas centrales que son Cabeceras las identifica con "1" y las que no son con "0" *)
hostx=Table[If[host1x[[ix]]==1,True,False],{ix,nx}];
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],hostx[[ix]],host1x[[ix]]},{ix,nx}],TableSpacing->{0,1},TableDirections->
{Column,Row},TableHeadings->{None,{"ix","idx","Nombre Central","hostx","host1x"}}]]];
```

■ Identificación de Centrales Host por Provincia (variable hostp)

```
hostp=zerop;
Do[If[hostx[[ix]],hostp[[provx[[ix]]]=ix];,{ix,nx}];
(* Identificamos el host por cada provincia *)
Print[TableForm[Table[{ip,provTXTp[[ip]],hostp[[ip]]},{ip,np}],TableSpacing->{0,1},TableDirections->
{Column,Row},TableHeadings->{None,{"ip","Nombre Provincia","hostp"}}]]];
```

■ Identificación de Conexiones entre centrales

■ Crea Variable Auxiliar de Conexiones (variable Conexion)

```
Conexion=Table[ix,{ix,nx}];
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],Conexion[[ix]],idx[[Conexion[[ix]]],y[[Conexion[[
ix]],13]]},{ix,nx}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre
Central","Conexion","id Conexion","Nombre Conexion"}}]]];
(* Se inicializa el vector de conexión (para saber a qué se refiere cada vector de conexión) la inicialización es
como si cada central se conectara consigo misma. *)
```

■ Crea Variable Auxiliar de Anillos de Cabeceras (variable AnilloCabeceras)

```
Print["AnilloCabeceras"];
AnilloCabeceras=zerop;
pos=1;
Do[If[host1x[[iax]]==1,
  AnilloCabeceras[[pos]]=iax;
  pos=pos+1];,{iax,nx}];
Print[TableForm[Table[{AnilloCabeceras[[pos]],If[AnilloCabeceras[[pos]]!=0,y[[AnilloCabeceras[[pos]],13]],""],provx[[
AnilloCabeceras[[pos]]],If[AnilloCabeceras[[pos]]!=0,provTXTp[[provx[[AnilloCabeceras[[pos]]]]],""]},{pos,np}],TableSpacing->
{0,1},TableHeadings->{None,{"pos","Nombre Central","Provincia","Nombre Provincia"}}]]];
(* Anillos Cabeceras, marca las cabeceras provinciales *)
```

■ Crea Variable Auxiliar de Conexión de Provincias con su Tandem (variable ConexionProv)

```
Print["ConexionProv"];
ConexionProv=Table[ip,{ip,np}];
Do[If[AnilloCabeceras[[ip]]!=0,
  ConexionProv[[provx[[AnilloCabeceras[[ip]]]]]=provx[[tandemx[[AnilloCabeceras[[ip]]]]];,{ip,np}];
Print[TableForm[Table[{ip,provTXTp[[ip]],ConexionProv[[ip]],ConexionProv[[ip]]},{ip,np}],TableSpacing->
{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre
Central","Conexion","idConexion","Nombre Conexion"}}]]];
(* Inicializamos ConexionProv, aquí vemos que indicamos que cada Provincia (en realidad cabecera) se conecta con
cada tandem *)
```

■ Calcula distancias de cada central a su tandem (variable disttandemx)

```
Print["disttandemx"]
disttandemx=Abs[Table[distance[ix,tandemx[[ix]]],{ix,nx}]];
Do[If[esAnillodx[[ix]]==NA ^ host1x[[ix]]==1 ^ lima1x[[ix]]==1 ^ tandem1x[[ix]]==0,
  disttandemx[[ix]]=disttandemx[[ix]]+distance[ix,idinvertx[[377]]];
  ConexionProv[[provx[[ix]]]=provx[[idinvertx[[377]]]]];,{ix,nx}];
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],NumberForm[disttandemx[[ix]],{17,10}]],{ix,nx}],TableAlignments->
Right,TableSpacing->{0,1},TableHeadings->{None,{"ix","idx","Nombre Central","disttandemx"}}]]];
(* Para cada central se estima la distancia a su Tandem. Algunas centrales en Lima también están conectadas a la
central de San Isidro (la número 377), por ello para dichas centrales se suma la distancia con esa central.
Dichas centrales cumplen con: Ser Host o cabecera, No forman parte del anillo departamental, pertenecen a Lima y
no son ninguna de las dos Tandem de Lima *)
```

Calculo de la distancia entre nodos (remotas)

■ Identificación de Centrales Remotas (variables remote1x, remotex)

```
remote1x=Table[If[Systemx[[ix]]==0,1,0],{ix,nx}];
(* Para aquellas centrales que son remotas las identifica con "1" y las que no son remotas con "0" *)
remotex=Table[If[remote1x[[ix]]==1,True,False],{ix,nx}];
(* Para aquellas centrales que son Remotas las identifica con "True" y las que no son Remotas con "False" *)
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],remotex[[ix]],remote1x[[ix]]},{ix,nx}],TableSpacing->{0,1},TableDirections->{Column,Row},TableHeadings->{None,{"ix","idx","Nombre Central","remotex","remote1x"}}]]];
```

■ Identificación de host para el Resto del Perú (variable rophostx)

```
Print["rophostx"]
rophostx=Table[tandemx[[ix]],{ix,nx}];
Do[If[remotex[[ix]] ^ ropx[[ix]],
  Do[If[hostx[[jx]] ^ provx[[jx]]==provx[[ix]],
    rophostx[[ix]]=jx];,{jx,nx}];];{ix,nx}];
Print[TableForm[Table[{ix,y[[ix,13]],rophostx[[ix]],y[[rophostx[[ix]],13]]},{ix,nx}],TableSpacing->{0,1},TableHeadings->{None,{"ix","Nombre Central","rophostx","Nomobre Host"}}]]];
(* Sola para las centrales de provincias: si es remota se especifica cuál es su central cabecera o Host, es decir, la cabecera de su provincia. En el caso de las host queda como dato su Tandem. Observese que para identificar la cabecera de una remota es necesario buscar para cada remota cuál es la central cabecera que pertenece a la misma provincia *)
```

Identificación de Central de Jerarquia Superior para todas las Centrales

```
Print["allhostx y disthostx"]
allhostx=Table[tandemx[[ix]],{ix,nx}];
Do[If[remotex[[ix]],
  Do[If[hostx[[jx]] ^ provx[[jx]]==provx[[ix]],
    allhostx[[ix]]=jx];,{jx,nx}];];{ix,nx}];
(* Para todo el Perú: si es remota se especifica cuál es su central cabecera o Host, es decir, la cabecera de su provincia. En el caso de las host queda como dato su Tandem . Observese que para identificar la cabecera de una remota es necesario buscar para cada remota cuál es la central cabecera que pertenece a la misma provincia *)
disthostx=Abs[Table[distance[ix,allhostx[[ix]]],{ix,nx}]];
(* Para cada central se estima la distancia a la cabecera de su provincia. En el caso de las que son cabecera se establece como dato la distancia a su Tandem. No olvidar que en la variable allHostx a las cabeceras se les puso como allHostx la Tandem de su departamento *)
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],allhostx[[ix]],y[[allhostx[[ix]],13]],NumberForm[disthostx[[ix]],{17,10}]],{ix,nx}],TableSpacing->{0,1},TableHeadings->{None,{"ix","idx","Nombre Central","allhostx","Nombre Host","disthostx"}}]]];
```

Calcula Factor de NoLinealidad para cada conexion

■ Define variable auxiliar de Factor de No Linealidad (variable factorNoLinealx)

```
Print["factorNoLinealx"]
factorNoLinealx=onex;
(*Este listado corresponde a los enlaces que siendo de Tecnología Urbana (Canalizada), son tramos inter urbanos*)
EntreCiudades = {26, 39, 44, 45, 68, 69, 70, 95, 135, 136, 154, 156, 157, 158, 159, 162, 165, 170, 186, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 220, 221, 225, 226, 229, 232, 233, 234, 235, 236, 255, 256, 273, 274, 277, 281, 282, 308, 309, 329, 330, 357, 358, 456, 457, 464, 465, 467, 469, 470, 475, 476, 486, 487, 489, 490, 501, 502, 503, 505, 506, 507, 508, 526, 577, 580, 593, 609, 610, 616};
Do[If[transtechx[[ix]]==Enterrado v transtechx[[ix]]==Aereo,
  factorNoLinealx[[ix]]=factorNoLinealNoUrban];
  If[transtechx[[ix]]==Urbana,
    factorNoLinealx[[ix]]=factorNoLinealUrban];
  If[transtechx[[ix]]==Radio,
    factorNoLinealx[[ix]]=factorNoLinealRadio];
  If[Count[EntreCiudades, ix] > 0,
    factorNoLinealx[[ix]]=factorNoLinealNoUrban];
  ,{ix, nx}];
Print[TableForm[Table[{ix,idx[[ix]],y[[ix,13]],NumberForm[factorNoLinealx[[ix]],{17,10}]],{ix,nx}],TableAlignments->Right,TableSpacing->{0,1},TableHeadings->{None,{"ix","idx","Nombre Central","factorNoLinealx"}}]]];
(* Para cada central se especifica cuál es el factor de ajuste por no linealidad dependiendo de la tecnología de transmisión que se usa en dicho nodo *)
```


Calculo de la distancia entre nodos (centrales y remotas) ajustadas por el Factor de No Linealidad

```

Print["disthostx"];
disthostx =  $\frac{\text{disthostx}}{\text{factorNoLinealx}}$ ;
(* corrige la variable dishostx por el factor de no linealidad que le corresponde *)

Print["disttandemx"];
disttandemx =  $\frac{\text{disttandemx}}{\text{factorNoLinealx}}$ ;
(* corrige la variable disttandemx por el factor de no linealidad que le corresponde *)

Print["repfibrax"];
repfibrax = zerox;
Do[
  If[host1x[[ix]] == 1,
    repfibrax[[ix]] = disttandemx[[ix]];
    Conexion[[ix]] = ix;,
    repfibrax[[ix]] = disthostx[[ix]];
    Conexion[[ix]] = allhostx[[ix]];
  ];
, {ix, nx}];
(* Distancia entre repetidoras de fibra:
  Como punto de partida para las remotas toma la variable disthostx y para las host la variable disttandemx *)
Print["Conexion"];
Print[TableForm[
  Table[{ix, idx[[ix]], y[[ix, 13]], NumberForm[disthostx[[ix]], {17, 10}], NumberForm[disttandemx[[ix]], {17, 10}],
    NumberForm[repfibrax[[ix]], {17, 10}], Conexion[[ix]], y[[Conexion[[ix]], 13]], {ix, nx}},
  TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None,
    {"ix", "idx", "Nombre Central", "disthostx", "disttandemx", "repfibrax", "Conexión", "Nombre Conexión"}}]];

```


Calcula el Tamaño de los Anillos (Sumatoria de distancias)

■ Tamaño de Anillo a Nivel Provincial

```
Print[Style["A NIVEL PROVINCIAL", Underlined, 16, Background -> Blue, White, Bold]],

(*Cada provincia es un anillo de Remotas-Cabeceras*)
(*Inicializacion de variables*)
TamAnillo = zerop;
CabAnillo = 1;
ProvAnillo = 1;
esAnillo = 0;
costeTerminacionesFibrax = zerox;
SumaCabecerax = zerox;
Print["TamAnillo"];
Do[If[host1x[[iax]] == 1,
  If[esAnillox[[iax]] == 1,
    CabAnillo = iax;          (*Cabecera del anillo*)
    ProvAnillo = provx[[CabAnillo]]; (*Provincia del anillo*)
    If[provx[[iax + 1]] == ProvAnillo & esAnillox[[iax + 1]] == 1,
      costeTerminacionesFibrax[[iax + 1]] = 2 * TxFibraCosteTerminacion, 0];
    (*Coste de las term del enlace extra de los anillos provinciales*)
    (*Tecnologia que usan las centrales pertenecientes al anillo*)
    esAnillo = 1;,
    esAnillo = 0;];];
If[esAnillo == 1,
  If[host1x[[iax + 1]] == 1 ∨ esAnillox[[iax + 1]] == 0,

    TamAnillo[[ProvAnillo]] = TamAnillo[[ProvAnillo]] + If[esAnillox[[iax]] == 0, 0, 1] *  $\frac{\text{Abs}[\text{distance}[\text{iax}, \text{CabAnillo}]]}{\text{factorNoLinealx}[\text{iax}]}$ ;

    If[esAnillox[[iax]] == 1, SumaCabecerax[[iax]] =  $\frac{\text{Abs}[\text{distance}[\text{iax}, \text{CabAnillo}]]}{\text{factorNoLinealx}[\text{iax}]}$ ;;

    repfibrax[[iax]] = repfibrax[[iax]] +  $\frac{\text{Abs}[\text{distance}[\text{iax}, \text{CabAnillo}]]}{\text{factorNoLinealx}[\text{iax}]}$ ;];

    Conexion[[iax]] = CabAnillo;;

    TamAnillo[[ProvAnillo]] = TamAnillo[[ProvAnillo]] +  $\frac{\text{Abs}[\text{distance}[\text{iax}, \text{iax} + 1]]}{\text{factorNoLinealx}[\text{iax}]}$ ;

    repfibrax[[iax + 1]] =  $\frac{\text{Abs}[\text{distance}[\text{iax}, \text{iax} + 1]]}{\text{factorNoLinealx}[\text{iax}]}$ ;

    Conexion[[iax]] = iax + 1;];]; {iax, nx - 1}];
Print[TableForm[Table[{ip, provTXTp[[ip]], TamAnillo[[ip]]}, {ip, np}], TableAlignments -> Right,
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"ip", "Nombre Provincia", "TamAnillo"}}]];
(* Se estima el tamaño del anillo de cada provincia. Para ello hay que asignar a cada nodo una
  distancia relevante y luego sumar dichas distancias. Primero se busca el final del anillo,
  entonces, evalua cada nodo que si es anillo, luego,
  si el nodo siguiente es host o no forma parte de un anillo entonces estas en un nodo que cierra un anillo,
  por lo que la distancia relevante es la distancia con su cabecera. En cualquier otro caso,
  la distancia relevante es la distancia con el nodo siguiente *)
```


■ Tamaño de Anillo a Nivel Departamental

```
Print[Style["A NIVEL DEPARTAMENTAL", Underlined, 16, Background → Blue, White, Bold]];

(*Inicializacion de variables*)
Print["TamAnillod"];
TamAnillod = zerod;
TandemAnillo = 0;
DepAnillo = 0;
(*Se extraen las cabeceras*)
esAnillo = 0;
Do[
  If[AnilloCabeceras[[iax]] ≠ 0,
    If[tandem1x[[AnilloCabeceras[[iax]]]] == 1,
      If[esAnillodx[[AnilloCabeceras[[iax]]]] ≠ NA,
        TandemAnillo = AnilloCabeceras[[iax]]; (*Tandem del anillo*)
        DepAnillo = deptx[[TandemAnillo]]; (*Departamento del anillo*)
        esAnillo = 1;,
        esAnillo = 0;];];
    If[esAnillo == 1,
      If[tandem1x[[AnilloCabeceras[[iax + 1]]]] == 1 ∨ esAnillodx[[AnilloCabeceras[[iax + 1]]]] == NA,
        TamAnillod[[DepAnillo]] = TamAnillod[[DepAnillo]] +
          If[esAnillodx[[AnilloCabeceras[[iax]]]] == NA, 0, 1] *  $\frac{\text{Abs}[\text{distance}[\text{AnilloCabeceras}[[iax]], \text{TandemAnillo}]]}{\text{factorNoLinealx}[[\text{AnilloCabeceras}[[iax]]]]}$ ;
        If[esAnillodx[[AnilloCabeceras[[iax]]]] ≠ NA,
          repfibrax[[AnilloCabeceras[[iax]]]] =
            repfibrax[[AnilloCabeceras[[iax]]]] +  $\frac{\text{Abs}[\text{distance}[\text{AnilloCabeceras}[[iax]], \text{TandemAnillo}]]}{\text{factorNoLinealx}[[\text{AnilloCabeceras}[[iax]]]]}$ ;];
        ConexionProv[[provx[[AnilloCabeceras[[iax]]]]]] = provx[[TandemAnillo]],
        TamAnillod[[DepAnillo]] =
          TamAnillod[[DepAnillo]] +  $\frac{\text{Abs}[\text{distance}[\text{AnilloCabeceras}[[iax]], \text{AnilloCabeceras}[[iax + 1]]]]}{\text{factorNoLinealx}[[\text{AnilloCabeceras}[[iax]]]]}$ ;
        repfibrax[[AnilloCabeceras[[iax + 1]]]] =  $\frac{\text{Abs}[\text{distance}[\text{AnilloCabeceras}[[iax]], \text{AnilloCabeceras}[[iax + 1]]]]}{\text{factorNoLinealx}[[\text{AnilloCabeceras}[[iax]]]]}$ ;
        ConexionProv[[provx[[AnilloCabeceras[[iax]]]]]] = provx[[AnilloCabeceras[[iax + 1]]]];];];, {iax, np - 1}];
Print[TableForm[Table[{id, deptTXTd[[id]], TamAnillod[[id]]}, {id, nd}], TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "TamAnillod"}}]];
(* Se estima el tamaño del anillo de cada departamento. Para ello hay que asignar a cada nodo una
  distancia relevante y luego sumar dichas distancias. Primero se busca el final del anillo, entonces,
  evalua cada host que si es anillo, luego, si el host siguiente es cabecera de anillo departamental
  (es Tandem) o no forma parte de un anillo departamental entonces estas en un nodo que cierra un anillo,
  por lo que la distancia relevante es la distancia con su Tandem. En cualquier otro caso,
  la distancia relevante es la distancia con el nodo cabecera siguiente *)
```

■ Eliminacion de Valores muy bajos en Tamaño de Anillos

```
Print[Style["ELIMINACIÓN DE VALORES MUY BAJOS EN TAMAÑO DE ANILLOS",
  White, 16, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0], Bold]];

Print["Calculos posteriores"];
Do[If[TamAnillop[[iap]] < 0.01, TamAnillop[[iap]] = 0];, {iap, np}];
Do[If[TamAnillod[[iad]] < 0.01, TamAnillod[[iad]] = 0];, {iad, nd}];
Print["FIN Calculos posteriores"];
Print["TamAnillop"];
Print[TableForm[Table[{ip, provTXTp[[ip]], TamAnillop[[ip]]}, {ip, np}], TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"ip", "Nombre Provincia", "TamAnillop"}}]];
Print["TamAnillod"];
Print[TableForm[Table[{id, deptTXTd[[id]], TamAnillod[[id]]}, {id, nd}], TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "TamAnillod"}}]];
(* Las distancias menores a 0.01 kilómetros las hace cero *)
```

ESTIMACIÓN DE NÚMERO DE LINEAS

```
Print[Style["ESTIMACIONES DE NÚMERO DE LÍNEAS",
  White, 16, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0], Bold]];
```

```
linesp = zerop;
linesd = zerod;
satlinesx = linesx * sat1x;
(* Se especifica el número de líneas de cada una de las centrales que usan tecnología de Tx satelital *)
remotelinesd = zerod;
tandemlinesd = zerod;
remotetandemlinesd = zerod;
satlinesd = zerod;
satlinesp = zerop;
interprovfracx = zerox;
Do[linesp[[provx[[ix]]]] = linesp[[provx[[ix]]]] + linesx[[ix]];
  linesd[[deptx[[ix]]]] = linesd[[deptx[[ix]]]] + linesx[[ix]];
  (* Número de Líneas que existe en cada provincia y
    departamento: Suma las líneas de los nodos que pertenecen a cada provincia y departamento respectivamente*)
  remotelinesd[[deptx[[ix]]]] = remotelinesd[[deptx[[ix]]]] + remote1x[[ix]] * linesx[[ix]];
  (* Total de líneas conectadas sólo a las centrales remotas: Información a nivel departamental*)
  tandemlinesd[[deptx[[ix]]]] = tandemlinesd[[deptx[[ix]]]] + tandem1x[[ix]] * linesx[[ix]];
  (* Total de líneas conectadas sólo a las centrales Tandem: Información a nivel departamental*)
  If[remotex[[ix]] & allhostx[[ix]] == tandemx[[ix]],
    remotetandemlinesd[[deptx[[ix]]]] = remotetandemlinesd[[deptx[[ix]]]] + linesx[[ix]];];
  (* Se identifica el número de líneas de las remotas de un departamento,
    pero solo de aquellas remotas cuya cabecera es a su vez la tandem del departamento *)
  satlinesd[[deptx[[ix]]]] = satlinesd[[deptx[[ix]]]] + satlinesx[[ix]];
  (* Se totaliza a nivel departamental el número de líneas que usan tecnología de Tx satelital *)
  satlinesp[[provx[[ix]]]] = satlinesp[[provx[[ix]]]] + satlinesx[[ix]]; , {ix, nx}];
(* Se totaliza a nivel provincial el número de líneas que usan tecnología de Tx satelital *)
Do[If[linesp[[provx[[ix]]]] == 0 ∨ linesd[[deptx[[ix]]]] == 0,

  interprovfracx[[ix]] = 0;; interprovfracx[[ix]] =  $1 - \left( \frac{\text{linesp}[[\text{provx}[[ix]]]]}{\text{linesd}[[\text{deptx}[[ix]]]]} \right)^{\frac{1}{\text{locb}}}$ ;]; , {ix, nx}];

hostlinesd = linesd - remotelinesd;
(* Total de líneas conectadas sólo a las centrales Cabeceras: Información a nivel departamental*)

Print["lines"];
lines = Plus @@ linesd;
Print[lines];
(* Total de líneas en la red *)
Print[TableForm[Table[{id, deptTXTd[[id]], linesd[[id]], remotelinesd[[id]],
  tandemlinesd[[id]], hostlinesd[[id]], remotetandemlinesd[[id]], satlinesd[[id]]}, {id, nd}],
  TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento",
  "linesd", "remotelinesd", "tandemlinesd", "hostlinesd", "remotetandemlinesd", "satlinesd"}}]]];
Print[TableForm[Table[{ip, provTXTp[[ip]], linesp[[ip]], satlinesp[[ip]]}, {ip, np}], TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Provincia", "linesp", "satlinesp"}}]]];
Print[TableForm[Table[{ix, y[[ix, 13]], linesx[[ix]], satlinesx[[ix]], interprovfracx[[ix]]}, {ix, nx}],
  TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Nombre Central", "linesx", "satlinesx", "interprovfracx"}}]]];
(* Factor de elasticidad estimado por cada central para ser empleado en la
  estimación de cada uno de los distintos tipos de tráfico*)
```

TRÁFICO EN HORA CARGADA

```
Print[Style["TRATAMIENTO DEL TRÁFICO EN LA HORA CARGADA",
  White, 16, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0], Bold]];

FactTrafico = 1.0;
Print["Trafico en Hora Cargada"];
TraficoRealHCx = ReadList["input_Trafico_HC.txt", {Real, Real, Real, Real, Real, Real}] * FactTrafico;
(* Print[TableForm[TraficoRealHCx, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {Automatic,
  {"Local Entrante", "Local Saliente", "LDN Entrante", "LDN Saliente", "LDI Entrante", "LDI Saliente"}}]] *)
(* Leyendo la información de tráfico del respectivo archivo txt *)
intradepinHC1x = Table[TraficoRealHCx[[idx[[ix]]][1]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico Intradepartamental de entrada. Le añade los minutos no tarifcados *)
intradepouthHC1x = Table[TraficoRealHCx[[idx[[ix]]][2]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico Intradepartamental de salida. Le añade los minutos no tarifcados *)
interprovinHC1x = intradepinHC1x * interprovfracx;
(* Se calcula el Trafico Interprovincial de entrada *)
interprovouthHC1x = intradepouthHC1x * interprovfracx;
(* Se calcula el Trafico Interprovincial de salida *)
locinHC1x = intradepinHC1x * (onex - interprovfracx);
(* Se calcula el Trafico Intraprovincial de entrada *)
locouthHC1x = intradepouthHC1x * (onex - interprovfracx);
(* Se calcula el Trafico Intraprovincial de salida *)
recldnacHC1x = Table[TraficoRealHCx[[idx[[ix]]][3]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico LDN ENTRANTE. Le añade los minutos no tarifcados *)
ldHC1x = Table[TraficoRealHCx[[idx[[ix]]][4]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico LDN SALIENTE. Le añade los minutos no tarifcados *)
intlHC1x = Table[TraficoRealHCx[[idx[[ix]]][5]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico LDI ENTRANTE. Le añade los minutos no tarifcados *)
intlouthHC1x = Table[TraficoRealHCx[[idx[[ix]]][6]], {ix, nx}] * (1 + trafAdicNoTarif);
(* Lectura del Tráfico LDI SALIENTE. Le añade los minutos no tarifcados *)
ldintradHC1x = interprovouthHC1x + interprovinHC1x;
(* Total Trafico Interprovincial: Interprovincial Entrante + Interprovincial Saliente *)
reclHC1x = recldnacHC1x;
(* Igual que recldnacHC1x: Tráfico LDN ENTRANTE *)
locHC1x =  $\frac{\text{locinHC1x} + \text{locouthHC1x}}{2}$ ;
(* Tráfco Intraprovincial Promedio: (Intraprovincial Entrante + Saliente)/2 *)

intraHC1x = Table[If[linesp[[provx[[ix]]]] == 0, 0, locHC1x[[ix]] *  $\left(\frac{\text{linesx}[[ix]]}{\text{linesp}[[provx[[ix]]]}\right)^{\frac{1}{\text{locb}}}$ ], {ix, nx}];

(* Estimación del Tráfico Intracentral *)
Print[TableForm[Table[{ix, y[[ix, 13]], intradepinHC1x[[ix]], intradepouthHC1x[[ix]], interprovinHC1x[[ix]],
  interprovouthHC1x[[ix]], locinHC1x[[ix]], locouthHC1x[[ix]], recldnacHC1x[[ix]], ldHC1x[[ix]], intlHC1x[[ix]],
  intlouthHC1x[[ix]], ldintradHC1x[[ix]], reclHC1x[[ix]], locHC1x[[ix]], intraHC1x[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "Intra Dep In", "Intra Dep Out",
    "Inter Prov In", "Inter Prov Out", "Intra Prov In", "Intra Prov Out", "LDN In", "LDN Out",
    "LDI IN", "LDI Out", "Inter Prov", "LDN In", "Intra Prov Prom", "Intra Central"}}]]];
```

```

Print[Style["TRATAMIENTO DEL TRÁFICO DE SEGUNDOS DE USO",
  White, 16, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0], Bold]];

Print["Trafico Minutos de Uso"];
TraficoRealMx = ReadList["input_Trafico_M.txt", {Real, Real, Real, Real, Real, Real}] * FactTrafico;
(*Print[TableForm[TraficoRealMx, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {Automatic,
  {"Local Entrante", "Local Saliente", "LDN Entrante", "LDN Saliente", "LDI Entrante", "LDI Saliente"}}]]*)
(* Leyendo la información de tráfico del respectivo archivo txt *)

intradepinMx = Table[TraficoRealMx[[idx[[ix]]][1]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico Intradepartamental de entrada *)

intradepoutMx = Table[TraficoRealMx[[idx[[ix]]][2]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico Intradepartamental de salida *)

interprovinMx = intradepinMx * interprovfracx;
(* Se calcula el Trafico Interprovincial de entrada *)

interprovoutMx = intradepoutMx * interprovfracx;
(* Se calcula el Trafico Interprovincial de salida *)

locinMx = intradepinMx * (onex - interprovfracx);
(*Se calcula el Trafico Intraprovincial de entrada*)

locoutMx = intradepoutMx * (onex - interprovfracx);
(* Se calcula el Trafico Intraprovincial de salida *)

recldnacMx = Table[TraficoRealMx[[idx[[ix]]][3]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico LDN ENTRANTE *)

ldMx = Table[TraficoRealMx[[idx[[ix]]][4]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico LDN SALIENTE *)

intlInMx = Table[TraficoRealMx[[idx[[ix]]][5]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico LDI ENTRANTE *)

intloutMx = Table[TraficoRealMx[[idx[[ix]]][6]] *  $\frac{12}{11}$ , {ix, nx}];
(* Lectura del Tráfico LDI SALIENTE *)

ldintradMx = interprovoutMx + interprovinMx;
(* Total Trafico Interprovincial: Interprovincial Entrante + Interprovincial Saliente *)

recldMx = recldnacMx;
(* Igual que recldnacMx: Tráfico LDN ENTRANTE *)

locMx =  $\frac{\text{locinMx} + \text{locoutMx}}{2}$ ;
(* Tráfico Intraprovincial Promedio: (Intraprovincial Entrante + Saliente)/2 *)

intraMx = Table[If[linesp[[provx[[ix]]]] == 0, 0, locMx[[ix]] *  $\left(\frac{\text{linesx}[[ix]]}{\text{linesp}[[provx[[ix]]]}\right)^{\frac{1}{\text{locb}}}$ ], {ix, nx}];
(* Estimación del Tráfico Intracentral *)

Print[TableForm[Table[{ix, y[[ix, 13]], intradepinMx[[ix]], intradepoutMx[[ix]],
  interprovinMx[[ix]], interprovoutMx[[ix]], locinMx[[ix]], locoutMx[[ix]], recldnacMx[[ix]], ldMx[[ix]],
  intlInMx[[ix]], intloutMx[[ix]], ldintradMx[[ix]], recldMx[[ix]], locMx[[ix]], intraMx[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "Intra Dep In", "Intra Dep Out",
    "Inter Prov In", "Inter Prov Out", "Intra Prov In", "Intra Prov Out", "LDN In", "LDN Out",
    "LDI IN", "LDI Out", "Inter Prov", "LDN In", "Intra Prov Prom", "Intra Central"}}]];

```

```

Print[Style["ANUALIZACIÓN DE MINUTOS DE USO",
  Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["Por Central"];

intrax = intraMx *  $\frac{12}{60}$ ;
(* Tráfico Intranodo en minutos de uso anual *)

locinx = locinMx *  $\frac{12}{60}$ ;
(* Tráfico Intraprovincial Entrante en Minutos de uso anual *)

locoutx = locoutMx *  $\frac{12}{60}$ ;
(* Tráfico Intraprovincial Saliente en Minutos de uso anual *)

interprovinx = interprovinMx *  $\frac{12}{60}$ ;
(* Tráfico Interprovincial Entrante en la Minutos de uso anual *)

interprovoutx = interprovoutMx *  $\frac{12}{60}$ ;
(* Tráfico Interprovincial Saliente en la Minutos de uso anual *)

ldintradx = ldintradMx *  $\frac{12}{60}$ ;
(* Total Trafico Interprovincial en minutos de uso

```

```

anual: Interprovincial Entrante + Interprovincial Saliente *)

recldx = recldMx *  $\frac{12}{60}$ ;

(* Igual que recldnacMx: Tráfico LDN ENTRANTE en minutos de uso anual *)

ldx = ldMx *  $\frac{12}{60}$ ;

(* Lectura del Tráfico LDN SALIENTE en Minutos de uso anual *)

intlinx = intlinMx *  $\frac{12}{60}$ ;

(* Lectura del Tráfico LDI ENTRANTE en minutos de uso anual *)

intloutx = intloutMx *  $\frac{12}{60}$ ;

(* Lectura del Tráfico LDI SALIENTE en minutos de uso anual *)

locx = locMx *  $\frac{12}{60}$ ;

(* Tráf.Intraprovincial Promedio en minutos de uso anual:  $\frac{\text{Intraprovincial Entrante} + \text{Saliente}}{2}$  *)

Print[TableForm[Table[{ix, y[[ix, 13]], intrax[[ix]], locinx[[ix]], locoutx[[ix]], interprovinx[[ix]],
  interprovoutx[[ix]], ldintradx[[ix]], recldx[[ix]], ldx[[ix]], intlinx[[ix]], intloutx[[ix]], locx[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "NombreCentral", "intrax", "locinx", "locoutx", "interprovinx",
    "interprovoutx", "ldintradx", "recldx", "ldx", "intlinx", "intloutx", "locx"}}]];

Print["Por Provincia"]
intrap = zerop;
locinp = zerop;
locoutp = zerop;
interprovinp = zerop;
interprovoutp = zerop;
ldintradp = zerop;
recldp = zerop;
ldp = zerop;
intlinp = zerop;
intloutp = zerop;
locp = zerop;
Do[intrap[[provx[[ix]]]] = intrap[[provx[[ix]]]] + intrax[[ix]];
  locinp[[provx[[ix]]]] = locinp[[provx[[ix]]]] + locinx[[ix]];
  locoutp[[provx[[ix]]]] = locoutp[[provx[[ix]]]] + locoutx[[ix]];
  interprovinp[[provx[[ix]]]] = interprovinp[[provx[[ix]]]] + interprovinx[[ix]];
  interprovoutp[[provx[[ix]]]] = interprovoutp[[provx[[ix]]]] + interprovoutx[[ix]];
  ldintradp[[provx[[ix]]]] = ldintradp[[provx[[ix]]]] + ldintradx[[ix]];
  recldp[[provx[[ix]]]] = recldp[[provx[[ix]]]] + recldx[[ix]];
  ldp[[provx[[ix]]]] = ldp[[provx[[ix]]]] + ldx[[ix]];
  intlinp[[provx[[ix]]]] = intlinp[[provx[[ix]]]] + intlinx[[ix]];
  intloutp[[provx[[ix]]]] = intloutp[[provx[[ix]]]] + intloutx[[ix]];
  locp[[provx[[ix]]]] = locp[[provx[[ix]]]] + locx[[ix]];; {ix, nx}];

(* Acumula por provincia el Tráfico Intranodo, Intraprovincial,
Intraprovincial, Interprovincial, Interprovincial,
Interprovincial (Interprovincial Entrante + Interprovincial Saliente), LDN ENTRANTE, LDN SALIENTE,
LDI ENTRANTE, LDI SALIENTE y Promedio Intra Provincial Saliente + Entrante en minutos de uso anual *)

Print[TableForm[Table[{ip, provTXTp[[ip]], intrap[[ip]], locinp[[ip]], locoutp[[ip]], interprovinp[[ip]],
  interprovoutp[[ip]], ldintradp[[ip]], recldp[[ip]], ldp[[ip]], intlinp[[ip]], intloutp[[ip]], locp[[ip]]}, {ip, np}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ip", "NombreProvincia", "intrap", "locinp", "locoutp", "interprovinp",
    "interprovoutp", "ldintradp", "recldp", "ldp", "intlinp", "intloutp", "locp"}}]];

Print["Por Departamento"];
intrad = zerod;
locind = zerod;
locoutd = zerod;
interprovind = zerod;
interprovoutd = zerod;
ldintradd = zerod;
recldd = zerod;
ldd = zerod;
intlind = zerod;
intloutd = zerod;
locd = zerod;
Do[intrad[[deptx[[ix]]]] = intrad[[deptx[[ix]]]] + intrax[[ix]];
  locind[[deptx[[ix]]]] = locind[[deptx[[ix]]]] + locinx[[ix]];
  locoutd[[deptx[[ix]]]] = locoutd[[deptx[[ix]]]] + locoutx[[ix]];
  interprovind[[deptx[[ix]]]] = interprovind[[deptx[[ix]]]] + interprovinx[[ix]];
  interprovoutd[[deptx[[ix]]]] = interprovoutd[[deptx[[ix]]]] + interprovoutx[[ix]];
  ldintradd[[deptx[[ix]]]] = ldintradd[[deptx[[ix]]]] + ldintradx[[ix]];
  recldd[[deptx[[ix]]]] = recldd[[deptx[[ix]]]] + recldx[[ix]];
  ldd[[deptx[[ix]]]] = ldd[[deptx[[ix]]]] + ldx[[ix]];
  intlind[[deptx[[ix]]]] = intlind[[deptx[[ix]]]] + intlinx[[ix]];
  intloutd[[deptx[[ix]]]] = intloutd[[deptx[[ix]]]] + intloutx[[ix]];
  locd[[deptx[[ix]]]] = locd[[deptx[[ix]]]] + locx[[ix]];; {ix, nx}];

(* Acumula por departamento el Tráfico Intranodo, Intraprovincial,

```

```
Intraprovincial, Interprovincial, Interprovincial,
Interprovincial (Interprovincial Entrante + Interprovincial Saliente), LDN ENTRANTE, LDN SALIENTE,
LDI ENTRANTE, LDI SALIENTE y Promedio Intra Provincial Saliente + Entrante en minutos de uso anual *)
Print[TableForm[Table[{id, deptTXTd[id], intrad[id], locind[id], locoutd[id], interprovind[id],
    interprovoutd[id], ldintradd[id], recldd[id], ldd[id], intlind[id], intloutd[id], locd[id]}, {id, nd}],
    TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
    TableHeadings → {None, {"id", "NombreDepartamento", "intrad", "locind", "locoutd", "interprovind",
        "interprovoutd", "ldintradd", "recldd", "ldd", "intlind", "intloutd", "locd"}}]]];
```

```
Print[
    Style["HORA CARGADA EN ERLANGS", Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
Print["Por Central"];
intraHCx =  $\frac{\text{intraHC1x}}{60}$ ;
(* Tráfico Intranodo en minutos de hora cargada *)
locinHCx =  $\frac{\text{locinHC1x}}{60}$ ;
(* Tráfico Intraprovincial Entrante en Minutos de hora cargada *)
locouthCx =  $\frac{\text{locouthC1x}}{60}$ ;
(* Tráfico Intraprovincial Saliente en Minutos de hora cargada *)
interprovinHCx =  $\frac{\text{interprovinHC1x}}{60}$ ;
(* Tráfico Interprovincial Entrante en la Minutos de hora cargada *)
interprovouthCx =  $\frac{\text{interprovouthC1x}}{60}$ ;
(* Tráfico Interprovincial Saliente en la Minutos de hora cargada *)
ldintradHCx =  $\frac{\text{ldintradHC1x}}{60}$ ;
(* Total Trafico Interprovincial en minutos de hora
cargada: Interprovincial Entrante + Interprovincial Saliente *)
reclldHCx =  $\frac{\text{reclldHC1x}}{60}$ ;
(* Igual que recldnacMx: Tráfico LDN ENTRANTE en minutos de hora cargada *)
ldHCx =  $\frac{\text{ldHC1x}}{60}$ ;
(* Lectura del Tráfico LDN SALIENTE en Minutos de hora cargada *)
intlinHCx =  $\frac{\text{intlinHC1x}}{60}$ ;
(* Lectura del Tráfico LDI ENTRANTE en minutos de hora cargada *)
intlouthCx =  $\frac{\text{intlouthC1x}}{60}$ ;
(* Lectura del Tráfico LDI SALIENTE en minutos de hora cargada *)
locHCx =  $\frac{\text{locHC1x}}{60}$ ;
(* Tráf.Intraprovincial Promedio en minutos de hora cargada: (Intraprovincial Entrante + Saliente)/2 *)
Print[TableForm[Table[{ix, y[ix, 13], intraHCx[ix], locinHCx[ix], locouthCx[ix], interprovinHCx[ix],
    interprovouthCx[ix], ldintradHCx[ix], reclldHCx[ix], ldHCx[ix], intlinHCx[ix], intlouthCx[ix], locHCx[ix]},
    {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
    TableHeadings → {None, {"ix", "NombreCentral", "intraHCx", "locinHCx", "locouthCx", "interprovinHCx",
        "interprovouthCx", "ldintradHCx", "reclldHCx", "ldHCx", "intlinHCx", "intlouthCx", "locHCx"}}]]];

Print["Por Provincia"]
intraHCp = zerop;
locinHCp = zerop;
locouthCp = zerop;
interprovinHCp = zerop;
interprovouthCp = zerop;
ldintradHCp = zerop;
reclldHCp = zerop;
ldHCp = zerop;
intlinHCp = zerop;
intlouthCp = zerop;
locHCp = zerop;
Do[intraHCp[provx[ix]] = intraHCp[provx[ix]] + intraHCx[ix];
    locinHCp[provx[ix]] = locinHCp[provx[ix]] + locinHCx[ix];
    locouthCp[provx[ix]] = locouthCp[provx[ix]] + locouthCx[ix];
    interprovinHCp[provx[ix]] = interprovinHCp[provx[ix]] + interprovinHCx[ix];
    interprovouthCp[provx[ix]] = interprovouthCp[provx[ix]] + interprovouthCx[ix];
    ldintradHCp[provx[ix]] = ldintradHCp[provx[ix]] + ldintradHCx[ix];
    reclldHCp[provx[ix]] = reclldHCp[provx[ix]] + reclldHCx[ix];
    ldHCp[provx[ix]] = ldHCp[provx[ix]] + ldHCx[ix];
    intlinHCp[provx[ix]] = intlinHCp[provx[ix]] + intlinHCx[ix];
    intlouthCp[provx[ix]] = intlouthCp[provx[ix]] + intlouthCx[ix];
    locHCp[provx[ix]] = locHCp[provx[ix]] + locHCx[ix];, {ix, nx}];
(*Acumula por provincia el Tráfico Intranodo, Intraprovincial,
Intraprovincial, Interprovincial, Interprovincial,
```



```
Interprovincial (Interprovincial Entrante+Interprovincial Saliente), LDN ENTRANTE, LDN SALIENTE,
LDI ENTRANTE, LDI SALIENTE y Promedio Intra Provincial Saliente+Entrante en minutos de hora cargada*)
Print[TableForm[
  Table[{ip, provTXTp[ip], intraHCp[ip], locinHCp[ip], locouthCp[ip], interprovinHCp[ip], interprovouthCp[ip],
    ldintradHCp[ip], recldHCp[ip], ldHCp[ip], intlinHCp[ip], intlouthCp[ip], locHCp[ip]}, {ip, np}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ip", "NombreProvincia", "intraHCp", "locinHCp", "locouthCp", "interprovinHCp",
    "interprovouthCp", "ldintradHCp", "recldHCp", "ldHCp", "intlinHCp", "intlouthCp", "locHCp"}}]];

Print["Por Departamento"];
intraHCd = zerod;
locinHCd = zerod;
locouthCd = zerod;
interprovinHCd = zerod;
interprovouthCd = zerod;
ldintradHCd = zerod;
recldHCd = zerod;
ldHCd = zerod;
intlinHCd = zerod;
intlouthCd = zerod;
locHCd = zerod;
Do[intraHCd[deptx[ix]] = intraHCd[deptx[ix]] + intraHCx[ix];
  locinHCd[deptx[ix]] = locinHCd[deptx[ix]] + locinHCx[ix];
  locouthCd[deptx[ix]] = locouthCd[deptx[ix]] + locouthCx[ix];
  interprovinHCd[deptx[ix]] = interprovinHCd[deptx[ix]] + interprovinHCx[ix];
  interprovouthCd[deptx[ix]] = interprovouthCd[deptx[ix]] + interprovouthCx[ix];
  ldintradHCd[deptx[ix]] = ldintradHCd[deptx[ix]] + ldintradHCx[ix];
  recldHCd[deptx[ix]] = recldHCd[deptx[ix]] + recldHCx[ix];
  ldHCd[deptx[ix]] = ldHCd[deptx[ix]] + ldHCx[ix];
  intlinHCd[deptx[ix]] = intlinHCd[deptx[ix]] + intlinHCx[ix];
  intlouthCd[deptx[ix]] = intlouthCd[deptx[ix]] + intlouthCx[ix];
  locHCd[deptx[ix]] = locHCd[deptx[ix]] + locHCx[ix];, {ix, nx}];
(*Acumula por departamento el Tráfico Intranodo, Intraprovincial,
Intraprovincial, Interprovincial, Interprovincial,
Interprovincial (Interprovincial Entrante+Interprovincial Saliente), LDN ENTRANTE, LDN SALIENTE,
LDI ENTRANTE, LDI SALIENTE y Promedio Intra Provincial Saliente+Entrante en minutos de hora cargada*)
Print[TableForm[Table[{id, deptTXTd[id], intraHCd[id], locinHCd[id], locouthCd[id], interprovinHCd[id],
  interprovouthCd[id], ldintradHCd[id], recldHCd[id], ldHCd[id], intlinHCd[id], intlouthCd[id], locHCd[id]},
  {id, nd}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "NombreDepartamento", "intraHCd", "locinHCd", "locouthCd", "interprovinHCd",
    "interprovouthCd", "ldintradHCd", "recldHCd", "ldHCd", "intlinHCd", "intlouthCd", "locHCd"}}]]];
```

ESTIMACION DE RUTAS DIRECTAS

```
Print[Style["ESTIMACIONES DE RUTAS DIRECTAS",
  Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

OpenRead["input_RutasDirectas.txt"];
RD = N[ReadList["input_RutasDirectas.txt", Number, RecordLists → True]];
(*Lectura del fichero de rutas directas*)

nrd = Length[RD];
(* Numero de registros en el fichero de rutas directas*)

If[consideraRD == 0, Do[RD[i, 3] = 0;, {i, nrd}];, Print["Se consideran Rutas Directas"]];
Print[Style["ESTIMACIONES A NIVEL LOCAL", Bold, 16, Underlined, White, Background → Blue]];
(*Tráfico de salida*)
locoutRDx = zerox;
restalocoutHCx = zerox;
restalocoutx = zerox;
Do[restalocoutHCx[idinvertx[RD[i, 1]]] =
  restalocoutHCx[idinvertx[RD[i, 1]]] - interprovouthCx[idinvertx[RD[i, 1]]] * RD[i, 3];
  restalocoutx[idinvertx[RD[i, 1]]] = restalocoutx[idinvertx[RD[i, 1]]] -
    interprovoutx[idinvertx[RD[i, 1]]] * RD[i, 3];, {i, nrd}];
(* Tráfico interprovincial saliente hacia las rutas directas en HC en
minutos: Se define como negativo pues se restará del dimensionamiento necesario *)
(* El tráfico interprovincial saliente en la HC en minutos se multiplica por el porcentaje
del tráfico total que sale de cada nodo que corresponde a tráfico en las rutas directas *)
ldintradHCx = ldintradHCx + restalocoutHCx;
(* al total del tráfico interprovincial en la HC (entrante+saliente)
le resta el tráfico de las rutas directas por tráfico saliente *)
(* Tráfico interprovincial saliente hacia las rutas directas en minutos de uso
anuales: Se define como negativo pues se restará del dimensionamiento necesario *)
(* El tráfico interprovincial saliente en minutos de uso anual se multiplica por el porcentaje
del tráfico total que sale de cada nodo que corresponde a tráfico en rutas directas *)
ldintradx = ldintradx + restalocoutx;
(* al total del tráfico interprovincial en minutos de uso anual
```

```

(entrante+saliente) le resta el tráfico de las rutas directas por tráfico saliente *)

Print["Tráfico de Salida"]; Print[
  TableForm[Table[{ix, y[ix, 13], restalocoutHCx[ix], ldintradHCx[ix], restalocoutx[ix], ldintradx[ix]}, {ix, nx}],
    TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
    TableHeadings → {None, {"ix", "Nombre Central", "restalocoutHCx", "ldintradHCx", "restalocoutx", "ldintradx"}}]];
(*Tráfico de entrada*)
locinRDx = Table[{0, 0, 0}, {ird, nrd}];
restalocinHCx = zerox;
restalocinx = zerox;

Do[locinRDx[i] = {idinvertx[RD[i, 1]], idinvertx[RD[i, 2]], interprovoutHCx[idinvertx[RD[i, 1]]] *  $\frac{RD[i, 3]}{60}$ };
  restalocinHCx[idinvertx[RD[i, 2]]] =
    restalocinHCx[idinvertx[RD[i, 2]]] - interprovoutHCx[idinvertx[RD[i, 1]]] * RD[i, 3];
  restalocinx[idinvertx[RD[i, 2]]] = restalocinx[idinvertx[RD[i, 2]]] - interprovoutx[idinvertx[RD[i, 1]]] * RD[i, 3];
  (*Lo mismo para los minutos de uso*), {i, nrd}];

(* Para cada central se identifican las centrales que le envían tráfico mediante las RD. Luego,
se toma la variable interprovoutHCx o interprovincial saliente en la HC en minutos de
dichas centrales y se multiplica por su respectivo % de tráfico saliente a través de las RD,
se divide entre 60 para tener Erlangs. Luego, para cada una de las centrales de destino se realiza
la sumatoria de los resultados estimados en las centrales que le envían tráfico por las RD *)
(* Tráfico interprovincial entrante de las rutas directas en HC en
minuto: Se define como negativo pues se restará del dimensionamiento necesario *)
(* Para cada central se identifican las centrales que le envían tráfico mediante las RD. Luego,
se toma la variable interprovoutHCx o interprovincial saliente de dichas centrales en minutos
de HC y se multiplica por su respectivo % de tráfico saliente a través de las RD. Luego,
para cada una de las centrales de destino se realiza la sumatoria de los
resultados estimados en las centrales que le envían tráfico por las RD *)

ldintradHCx = ldintradHCx + restalocinHCx;
(* al total del tráfico interprovincial en minutos de HC (entrante+saliente)
se le resta ahora el tráfico de las rutas directas por tráfico entrante*)
(* Tráfico interprovincial entrante de las rutas directas en minutos de uso
anuales: Se define como negativo pues se restará del dimensionamiento necesario *)
(* Para cada central se identifican las centrales que le envían tráfico mediante las RD. Luego,
se toma la variable interprovoutx o interprovincial saliente anual de dichas centrales en
minutos de uso y se multiplica por su respectivo % de tráfico saliente a través de las RD. Luego,
para cada una de las centrales de destino se realiza la sumatoria de los
resultados estimados en las centrales que le envían tráfico por las RD *)

ldintradx = ldintradx + restalocinx;
(* al total del tráfico interprovincial en minutos de uso anual
(entrante+saliente) le resta ahora el tráfico de las rutas directas por tráfico entrante *)

loadRDx = locinRDx;
(* Igual a la variable locinRDx equivalente al total de las cargas en
Erlangs que entran a cada central a través de RDS (tráfico interprovincial). Recordar
que: para cada central se identifican las centrales que le envían tráfico mediante las RD. Luego,
se toma la variable interprovoutHCx o interprovincial saliente en minutos de HC de dichas
centrales y se multiplica por su respectivo % de tráfico saliente a través de las RD,
se divide entre 60 para tener Erlangs. Luego, para cada una de las centrales de destino se realiza
la sumatoria de los resultados estimados en las centrales que le envían tráfico por las RD *)

Print["Tráfico de Entrada"]; Print[
  TableForm[Table[{ix, y[ix, 13], restalocinHCx[ix], ldintradHCx[ix], restalocinx[ix], ldintradx[ix]}, {ix, nx}],
    TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix",
      "Nombre Central", "restalocinHCx", "ldintradHCx actualizado", "restalocinx", "ldintradx actualizado"}}]];

Print["Rutas Directas"]; Print[
  TableForm[Table[{loadRDx[ird, 1], y[loadRDx[ird, 1], 13], loadRDx[ird, 2], y[loadRDx[ird, 2], 13], loadRDx[ird, 3]},
    {ird, nrd}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
    TableHeadings → {None, {"ix", "Nombre Central Origen", "ix", "Nombre Central Origen", "loadRDx"}}]];

Print[Style["ESTIMACIONES A NIVEL LDN", Bold, 16, Underlined, White, Background → Blue]];
reclimaldx = recldx lima1x;
(* Tráfico LDN ENTRANTE en minutos de uso anual en los nodos de Lima *)
reclimaldHCx = recldHCx lima1x;
(* Tráfico LDN ENTRANTE en HC anual en los nodos de Lima *)
recropinterldx = recldx rop1x;
(* Tráfico LDN ENTRANTE en minutos de uso anual en los nodos de provincias *)
recropinterldHCx = recldHCx rop1x;
(* Tráfico LDN ENTRANTE en HC anual en los nodos de provincias *)
Print[TableForm[Table[{ix, y[ix, 13], reclimaldx[ix], reclimaldHCx[ix], recropinterldx[ix], recropinterldHCx[ix]},
  {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Nombre Central", "reclimaldx", "reclimaldHCx", "recropinterldx", "recropinterldHCx"}}]];

```



```
Print[Style["RESUMEN DE TRÁFICO CONMUTABLE",
  Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
```

```
Print[Style["MINUTOS DE USO", Bold, 16, Underlined, White, Background → Blue]];
```

```
switchendLocx = locinx + locoutx + ldintradx - intrax;
(* Tráfico Local que conmuta en minutos de uso anual,
se suman entonces los tráficos: Intraprovincial entrante + Intraprovincial saliente + Interprovincial Total
- El tráfico intranodo (se resta porque esta duplicado en el intraprovincial) *)
(* Ojo que el tráficos de Interprovincial Total ya esta descontado el tráfico de las Rutas Directas *)
switchendLDNx = ldx + reclimaldx + recropinterldx;
(* Tráfico LDN que conmuta en minutos de uso anual, se suman entonces los
tráficos: LDN saliente + LDN Entrante en los nodos de Lima + LDN Entrante en los nodos de Provincias *)
(* Ojo que los tráficos de LDN Entrante en los nodos de Lima y LDN Entrante en los
nodos de Provincias ya estan descontados del tráfico de las Rutas Directas *)
switchendLDIx = intloutx + intltx;
(* Tráfico LDI que conmuta en minutos de uso anual,
se suman entonces los tráficos: LDI Saliente + LDI Entrante *)
switchendx = switchendLocx + switchendLDNx + switchendLDIx;
(* Continuación de Switchendx *)
TempLoc = 0;
TempLDN = 0;
TempLDI = 0;
Do[If[remotex[[ix]] ∧ tandemx[[ix]] == idinvertx[[248]],
  TempLoc = TempLoc + switchendLocx[[ix]];
  TempLDN = TempLDN + switchendLDNx[[ix]];
  TempLDI = TempLDI + switchendLDIx[[ix]];], {ix, nx}];
Do[If[hostx[[ix]] ∧ tandemx[[ix]] ≠ ix ∧ tandemx[[ix]] == idinvertx[[248]],
  TempLoc = TempLoc + switchendLocx[[ix]];
  TempLDN = TempLDN + switchendLDNx[[ix]];
  TempLDI = TempLDI + switchendLDIx[[ix]];], {ix, nx}];
switchendLocx[[248]] =  $\frac{\text{TempLoc}}{\text{TempLoc} + \text{TempLDN} + \text{TempLDI}}$ ;
switchendLDNx[[248]] =  $\frac{\text{TempLDN}}{\text{TempLoc} + \text{TempLDN} + \text{TempLDI}}$ ;
switchendLDIx[[248]] =  $\frac{\text{TempLDI}}{\text{TempLoc} + \text{TempLDN} + \text{TempLDI}}$ ;
(* Tráfico total que conmuta en minutos de uso anual,
se suman entonces los tráficos: switchendLocx + switchendLDNx + switchendLDIx *)
(* Continúa al final de la sección *)
Print[TableForm[Table[{ix, y[[ix, 13]], switchendLocx[[ix]], switchendLDNx[[ix]], switchendLDIx[[ix]], switchendx[[ix]]},
  {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Nombre Central", "switchendLocx", "switchendLDNx", "switchendLDIx", "switchendx"}}]]];

switchendLocp = zerop;
switchendLDNp = zerop;
switchendLDIp = zerop;
Do[switchendLocp[[provx[[ix]]]] = switchendLocp[[provx[[ix]]]] + switchendLocx[[ix]];
  switchendLDNp[[provx[[ix]]]] = switchendLDNp[[provx[[ix]]]] + switchendLDNx[[ix]];
  switchendLDIp[[provx[[ix]]]] = switchendLDIp[[provx[[ix]]]] + switchendLDIx[[ix]];], {ix, nx}];
switchendp = switchendLocp + switchendLDNp + switchendLDIp;
Print[TableForm[
  Table[{ip, provTXTp[[ip]], switchendLocp[[ip]], switchendLDNp[[ip]], switchendLDIp[[ip]], switchendp[[ip]]}, {ip, np}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ip", "Nombre Provincia", "switchendLocp", "switchendLDNp", "switchendLDIp", "switchendp"}}]]];
(* Se acumula a nivel provincial el trafico Local, LDN, LDI y Total que conmuta en minutos de uso anual *)
switchendLocd = zerod;
switchendLDNd = zerod;
switchendLDId = zerod;
Do[switchendLocd[[deptx[[ix]]]] = switchendLocd[[deptx[[ix]]]] + switchendLocx[[ix]];
  switchendLDNd[[deptx[[ix]]]] = switchendLDNd[[deptx[[ix]]]] + switchendLDNx[[ix]];
  switchendLDId[[deptx[[ix]]]] = switchendLDId[[deptx[[ix]]]] + switchendLDIx[[ix]];], {ix, nx}];
switchendd = switchendLocd + switchendLDNd + switchendLDId;
Print[TableForm[
  Table[{id, deptTXTd[[id]], switchendLocd[[id]], switchendLDNd[[id]], switchendLDId[[id]], switchendd[[id]]}, {id, nd}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Nombre Departamento", "switchendLocd", "switchendLDNd", "switchendLDId", "switchendd"}}]]];
(* Se acumula a nivel departamental el trafico Local, LDN, LDI y Total que conmuta en minutos de uso anual *)
```

```
Print[Style["HORA CARGADA", Bold, 16, Underlined, White, Background → Blue]];
```

```
switchendHCLocx = locinHCx + locoutHCx + ldintradHCx - intraHCx;
(* Tráfico Local que conmuta en minutos de HC,
se suman entonces los tráficos: Intraprovincial entrante + Intraprovincial saliente + Interprovincial Total
- El tráfico intranodo (se resta porque esta duplicado en el intraprovincial) *)
(* Ojo que el tráfico Interprovincial Total ya esta descontado del tráfico de las Rutas Directas *)
switchendHCLDNx = ldHCx + reclimaldHCx + recropinterldHCx;
(* Tráfico LDN que conmuta en minutos de HC, se suman entonces los
```

```
tráficos: LDN saliente + LDN Entrante en los nodos de Lima + LDN Entrante en los nodos de Provincias *)
(* Ojo que los tráfico LDN Entrante en los nodos de Lima y LDN Entrante en los
nodos de Provincias ya estan descontados del tráfico de las Rutas Directas *)
switchendHCLDIx = intlouthCx + intlincCx;
(* Tráfico LDI que conmuta en minutos de HC, se suman entonces los tráfico: LDI Saliente + LDI Entrante *)
switchendHCx = switchendHCLocx + switchendHCLDNx + switchendHCLDIx;
(* Tráfico total que conmuta en minutos de HC,
se suman entonces los tráfico: switchendHCLocx+switchendHCLDNx+switchendHCLDIx *)
Print[
  TableForm[Table[{ix, y[ix, 13], switchendHCLocx[ix], switchendHCLDNx[ix], switchendHCLDIx[ix], switchendHCx[ix]},
    {ix, nx}], TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings ->
    {None, {"ix", "Nombre Central", "switchendHCLocx", "switchendHCLDNx", "switchendHCLDIx", "switchendHCx"}}]];

switchendHCLocp = zerop;
switchendHCLDNp = zerop;
switchendHCLDIp = zerop;
Do[switchendHCLocp[provx[ix]] = switchendHCLocp[provx[ix]] + switchendHCLocx[ix];
  switchendHCLDNp[provx[ix]] = switchendHCLDNp[provx[ix]] + switchendHCLDNx[ix];
  switchendHCLDIp[provx[ix]] = switchendHCLDIp[provx[ix]] + switchendHCLDIx[ix];, {ix, nx}];
switchendHCp = switchendHCLocp + switchendHCLDNp + switchendHCLDIp;
Print[TableForm[Table[
  {ip, provTXTp[ip], switchendLocp[ip], switchendHCLDNp[ip], switchendHCLDIp[ip], switchendHCp[ip]}, {ip, np}],
  TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings ->
  {None, {"ip", "Nombre Provincia", "switchendHCLocp", "switchendHCLDNp", "switchendHCLDIp", "switchendHCp"}}]];
(* Se acumula a nivel provincial el trafico Local, LDN, LDI y Total que conmuta en minutos de HC *)

switchendHCLocd = zerod;
switchendHCLDNd = zerod;
switchendHCLDIId = zerod;
Do[switchendHCLocd[deptx[ix]] = switchendHCLocd[deptx[ix]] + switchendHCLocx[ix];
  switchendHCLDNd[deptx[ix]] = switchendHCLDNd[deptx[ix]] + switchendHCLDNx[ix];
  switchendHCLDIId[deptx[ix]] = switchendHCLDIId[deptx[ix]] + switchendHCLDIx[ix];, {ix, nx}];
switchendHCd = switchendHCLocd + switchendHCLDNd + switchendHCLDIId;
Print[TableForm[Table[
  {id, deptXTD[id], switchendHCLocd[id], switchendHCLDNd[id], switchendHCLDIId[id], switchendHCd[id]}, {id, nd}],
  TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings -> {None,
  {"id", "Nombre Departamento", "switchendHCLocd", "switchendHCLDNd", "switchendHCLDIId", "switchendHCd"}}]];
(* Se acumula a nivel departamental el trafico Local, LDN, LDI y Total que conmuta en minutos de HC *)
```

```
Print[Style["IDENTIFICACIÓN DE CARGAS EN ERLANGS",
  Bold, 16, Underlined, White, Background -> RGBColor[0, 1/2, 0]]];

Print[Style["CARGAS SATELITALES", Bold, 16, Underlined, White, Background -> Blue]];
Print[Style["MINUTOS DE USO", Bold, 16, Underlined, White, Background -> Blue]];

satminLocx = switchendLocx sat1x;
satminLDNx = switchendLDNx sat1x;
satminLDIx = switchendLDIx sat1x;
satldintradx = ldintradx sat1x;
satminx = (satminLocx + satminLDNx + satminLDIx) sat1x;
Print[TableForm[
  Table[{ix, y[ix, 13], satminLocx[ix], satminLDNx[ix], satminLDIx[ix], satldintradx[ix], satminx[ix]}, {ix, nx}],
  TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings ->
  {None, {"ix", "Nombre Central", "satminLocx", "satminLDNx", "satminLDIx", "satldintradx", "switchendx"}}]];

satminLocp = zerop;
satminLDNp = zerop;
satminLDIp = zerop;
satldintradp = zerop;
satminp = zerop;
Do[satminLocp[provx[ix]] = satminLocp[provx[ix]] + satminLocx[ix];
  satminLDNp[provx[ix]] = satminLDNp[provx[ix]] + satminLDNx[ix];
  satminLDIp[provx[ix]] = satminLDIp[provx[ix]] + satminLDIx[ix];
  satldintradp[provx[ix]] = satldintradp[provx[ix]] + satldintradx[ix];
  satminp[provx[ix]] = satminp[provx[ix]] + satminx[ix];, {ix, nx}];
Print[TableForm[Table[
  {ip, provTXTp[ip], satminLocp[ip], satminLDNp[ip], satminLDIp[ip], satldintradp[ip], satminp[ip]}, {ip, np}],
  TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings ->
  {None, {"ip", "Nombre Provincia", "satminLocp", "satminLDNp", "satminLDIp", "satldintradp", "switchendp"}}]];

satminLocd = zerod;
satminLDNd = zerod;
satminLDId = zerod;
satldintradd = zerod;
satmind = zerod;
Do[satminLocd[deptx[ix]] = satminLocd[deptx[ix]] + satminLocx[ix];
  satminLDNd[deptx[ix]] = satminLDNd[deptx[ix]] + satminLDNx[ix];
  satminLDId[deptx[ix]] = satminLDId[deptx[ix]] + satminLDIx[ix];
```

```

satldintradd[depts[ix]] = satldintradd[depts[ix]] + satldintradx[ix];
satmind[depts[ix]] = satmind[depts[ix]] + satminx[ix], {ix, nx}];
Print[TableForm[Table[
  {id, deptTXTd[id], satminLocd[id], satminLDNd[id], satminLDId[id], satldintradd[id], satmind[id]}, {id, nd}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None,
    {"id", "Nombre Departamento", "satminLocd", "satminLDNd", "satminLDId", "satldintradd", "switchendd"}}]];

Print[Style["HORA CARGADA", Bold, 16, Underlined, White, Background → Blue]],
satminHCLocx = switchendHCLocx sat1x;
satminHCLDNx = switchendHCLDNx sat1x;
satminHCLDIX = switchendHCLDIX sat1x;
satldintradHCx = ldintradHCx sat1x;
satminHCx = (satldintradHCx + satminHCLDNx + satminHCLDIX) * sat1x;
Print[TableForm[
  Table[{ix, y[ix, 13], satminHCLocx[ix], satminHCLDNx[ix], satminHCLDIX[ix], satldintradHCx[ix], satminHCx[ix]},
    {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None,
    {"ix", "Nombre Central", "satminHCLocx", "satminHCLDNx", "satminHCLDIX", "satldintradHCx", "switchendHCx"}}]];
(* Total tráfico en minutos de HC que puede conmutar a nivel de
Tx. satelital: Tráfico total interprovincial (entrante+saliente)+ LDN Saliente + LDN Entrante + LDI Saliente+
LDI Entrante. Ojo que al tráfico interprovincial total ya se le descontó el tráfico de las rutas directas *)

loadsatLocx =  $\frac{\text{satminHCLocx}}{60}$ ;
loadsatLDNx =  $\frac{\text{satminHCLDNx}}{60}$ ;
loadsatLDIX =  $\frac{\text{satminHCLDIX}}{60}$ ;
loadsatx =  $\frac{\text{satminHCx}}{60}$ ;
Print[TableForm[Table[{ix, y[ix, 13], loadsatLocx[ix], loadsatLDNx[ix], loadsatLDIX[ix], loadsatx[ix]}, {ix, nx}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "loadsatLocx", "loadsatLDNx", "loadsatLDIX", "loadsatx"}}]];
(* Se divide el tráfico considerado para el dimensionamiento satelital entre 60 para expresarlo en Erlangs *)
(* Tablas Ponderadoras *)

satLocx = Table[If[loadsatx[ix] > 0,  $\frac{\text{loadsatLocx[ix]}}{\text{loadsatx[ix]}}$ , 0], {ix, nx}];
satLDNx = Table[If[loadsatx[ix] > 0,  $\frac{\text{loadsatLDNx[ix]}}{\text{loadsatx[ix]}}$ , 0], {ix, nx}];
satLDIX = Table[If[loadsatx[ix] > 0,  $\frac{\text{loadsatLDIX[ix]}}{\text{loadsatx[ix]}}$ , 0], {ix, nx}];

Print[Style["TRAMO REMOTA-CABECERA", Bold, 16, Underlined, White, Background → Blue]],
ropminremotehostLocx = Table[If[remotex[ix] ∧ ropx[ix], switchendLocx[ix] - intrax[ix], 0], {ix, nx}];
ropminremotehostLDNx = Table[If[remotex[ix] ∧ ropx[ix], switchendLDNx[ix], 0], {ix, nx}];
ropminremotehostLDIX = Table[If[remotex[ix] ∧ ropx[ix], switchendLDIX[ix], 0], {ix, nx}];
ropminremotehostx = ropminremotehostLocx + ropminremotehostLDNx + ropminremotehostLDIX;
Print[
  TableForm[Table[{ix, y[ix, 13], ropminremotehostLocx[ix], ropminremotehostLDNx[ix], ropminremotehostLDIX[ix],
    ropminremotehostx[ix]}, {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central", "ropminremotehostLocx",
    "ropminremotehostLDNx", "ropminremotehostLDIX", "ropminremotehostx"}}]];

ropminremotehostHCLocx = Table[If[remotex[ix] ∧ ropx[ix], switchendHCLocx[ix] - intraHCx[ix], 0], {ix, nx}];
ropminremotehostHCLDNx = Table[If[remotex[ix] ∧ ropx[ix], switchendHCLDNx[ix], 0], {ix, nx}];
ropminremotehostHCLDIX = Table[If[remotex[ix] ∧ ropx[ix], switchendHCLDIX[ix], 0], {ix, nx}];
ropminremotehostHCx = ropminremotehostHCLocx + ropminremotehostHCLDNx + ropminremotehostHCLDIX;
Print[TableForm[Table[{ix, y[ix, 13], ropminremotehostHCLocx[ix],
  ropminremotehostHCLDNx[ix], ropminremotehostHCLDIX[ix], ropminremotehostHCx[ix]}, {ix, nx}],
  TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "ropminremotehostHCLocx",
    "ropminremotehostHCLDNx", "ropminremotehostHCLDIX", "ropminremotehostHCx"}}]];
(* Para las centrales remotas y de provincias se toma el tráfico total considerado como tráfico
que conmuta y se le resta el tráfico intranodo pues no pasa por la jerarquía R-H *)

limaminremotehostLocx = (switchendLocx - intrax) (lima1x * remote1x);
limaminremotehostLDNx = (switchendLDNx) (lima1x * remote1x);
limaminremotehostLDIX = (switchendLDIX) (lima1x * remote1x);
limaminremotehostx = limaminremotehostLocx + limaminremotehostLDNx + limaminremotehostLDIX;
Print[
  TableForm[Table[{ix, y[ix, 13], limaminremotehostLocx[ix], limaminremotehostLDNx[ix], limaminremotehostLDIX[ix],
    limaminremotehostx[ix]}, {ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right,
  TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central", "limaminremotehostLocx",
    "limaminremotehostLDNx", "limaminremotehostLDIX", "limaminremotehostx"}}]];

limaminremotehostHCLocx = (switchendHCLocx - intraHCx) * (lima1x * remote1x);
limaminremotehostHCLDNx = (switchendHCLDNx) * (lima1x * remote1x);
limaminremotehostHCLDIX = (switchendHCLDIX) * (lima1x * remote1x);
limaminremotehostHCx = limaminremotehostHCLocx + limaminremotehostHCLDNx + limaminremotehostHCLDIX;
Print[TableForm[Table[{ix, y[ix, 13], limaminremotehostHCLocx[ix],

```

```

limaminremotehostHCLDNx[[ix]], limaminremotehostHCLDIx[[ix]], limaminremotehostHCx[[ix]], {ix, nx}},
TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
TableHeadings → {None, {"ix", "Nombre Central", "limaminremotehostHCLocx",
"limaminremotehostHCLDNx", "limaminremotehostHCLDIx", "limaminremotehostHCx"}}]]];
(* Para las centrales remotas y de Lima se toma el tráfico total considerado como tráfico
que conmuta y se le resta el tráfico intranodo pues no pasa por la jerarquía R-H *)
minremotehostLocx = rop1x ropminremotehostLocx + lima1x limaminremotehostLocx;
minremotehostLDNx = rop1x ropminremotehostLDNx + lima1x limaminremotehostLDNx;
minremotehostLDIx = rop1x ropminremotehostLDIx + lima1x limaminremotehostLDIx;
minremotehostx = rop1x ropminremotehostx + lima1x limaminremotehostx;
Print[TableForm[
Table[{ix, y[[ix, 13]], minremotehostLocx[[ix]], minremotehostLDNx[[ix]], minremotehostLDIx[[ix]], minremotehostx[[ix]],
{ix, nx}}, TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None,
{"ix", "Nombre Central", "minremotehostLocx", "minremotehostLDNx", "minremotehostLDIx", "minremotehostx"}}]]];

minremotehostLocd = zerod;
minremotehostLDNd = zerod;
minremotehostLDId = zerod;
minremotehostd = zerod;
Do[minremotehostLocd[[deptx[[ix]]]] = minremotehostLocd[[deptx[[ix]]]] + minremotehostLocx[[ix]];
minremotehostLDNd[[deptx[[ix]]]] = minremotehostLDNd[[deptx[[ix]]]] + minremotehostLDNx[[ix]];
minremotehostLDId[[deptx[[ix]]]] = minremotehostLDId[[deptx[[ix]]]] + minremotehostLDIx[[ix]];
minremotehostd[[deptx[[ix]]]] = minremotehostd[[deptx[[ix]]]] + minremotehostx[[ix]], {ix, nx}];
Print[TableForm[Table[{id, deptXTd[id]], minremotehostLocd[id], minremotehostLDNd[id],
minremotehostLDId[id], minremotehostd[id]], {id, nd}], TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento",
"minremotehostLocd", "minremotehostLDNd", "minremotehostLDId", "minremotehostd"}}]]];

minremotehostHCLocx = rop1x * ropminremotehostHCLocx + lima1x * limaminremotehostHCLocx;
minremotehostHCLDNx = rop1x * ropminremotehostHCLDNx + lima1x * limaminremotehostHCLDNx;
minremotehostHCLDIx = rop1x * ropminremotehostHCLDIx + lima1x * limaminremotehostHCLDIx;
minremotehostHCx = rop1x * ropminremotehostHCx + lima1x * limaminremotehostHCx;
Print[TableForm[Table[{ix, y[[ix, 13]], minremotehostHCLocx[[ix]], minremotehostHCLDNx[[ix]],
minremotehostHCLDIx[[ix]], minremotehostHCx[[ix]], {ix, nx}}, TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central",
"minremotehostHCLocx", "minremotehostHCLDNx", "minremotehostHCLDIx", "minremotehostHCx"}}]]];
(* Para Todas las centrales remotas se toma el tráfico total considerado como tráfico que
conmuta y se le resta el tráfico intranodo pues no pasa por la jerarquía R-H *)

loadremotehostLocx =  $\frac{\text{minremotehostHCLocx}}{60}$ ;
loadremotehostLDNx =  $\frac{\text{minremotehostHCLDNx}}{60}$ ;
loadremotehostLDIx =  $\frac{\text{minremotehostHCLDIx}}{60}$ ;
loadremotehostx =  $\frac{\text{minremotehostHCx}}{60}$ ;
Print[TableForm[Table[{ix, y[[ix, 13]], loadremotehostLocx[[ix]], loadremotehostLDNx[[ix]],
loadremotehostLDIx[[ix]], loadremotehostx[[ix]], {ix, nx}}, TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central",
"loadremotehostLocx", "loadremotehostLDNx", "loadremotehostLDIx", "loadremotehostx"}}]]];
(* Para todas las remotas se divide el tráfico conmutable Local, LDN,
LDI y TOTAL entre 60 para identificar la carga correspondiente en Erlangs en el nivel R-H *)

Print[Style["TRAMO CABECERA-TANDEM", Bold, 16, Underlined, White, Background → Blue]];
(* A partir de aquí se retira la condición que excluye los satelitales *)
ropminhosttandemLocx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix (* ∧ ~satx[[ix]] *),
switchendLocp[[provx[[ix]]]] - (locinp[[provx[[ix]]]] + locoutp[[provx[[ix]]]] - intrap[[provx[[ix]]]]), 0] +
If[satx[[ix]], intrap[[provx[[ix]]]], 0], {ix, nx}];
ropminhosttandemLDNx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix
(* ∧ ~satx[[ix]] *), switchendLDNp[[provx[[ix]]]], 0], {ix, nx}];
ropminhosttandemLDIx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix
(* ∧ ~satx[[ix]] *), switchendLDIp[[provx[[ix]]]], 0], {ix, nx}];
ropminhosttandemx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix (* ∧ ~satx[[ix]] *),
switchendp[[provx[[ix]]]] - (locinp[[provx[[ix]]]] + locoutp[[provx[[ix]]]] - intrap[[provx[[ix]]]]), 0] +
If[satx[[ix]], intrap[[provx[[ix]]]], 0], {ix, nx}];
Print[TableForm[Table[{ix, y[[ix, 13]], ropminhosttandemLocx[[ix]], ropminhosttandemLDNx[[ix]],
ropminhosttandemLDIx[[ix]], ropminhosttandemx[[ix]], {ix, nx}}, TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central",
"ropminhosttandemLocx", "ropminhosttandemLDNx", "ropminhosttandemLDIx", "ropminhosttandemx"}}]]];

ropminhosttandemHCLocx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix (* ∧ ~satx[[ix]] *),
switchendHCLocp[[provx[[ix]]]] - (locinpHCP[[provx[[ix]]]] + locoutpHCP[[provx[[ix]]]] - intraHCP[[provx[[ix]]]]) +
If[satx[[ix]], intraHCP[[provx[[ix]]]], 0], 0], {ix, nx}];
ropminhosttandemHCLDNx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix
(* ∧ ~satx[[ix]] *), switchendHCLDNp[[provx[[ix]]]], 0], {ix, nx}];
ropminhosttandemHCLDIx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix
(* ∧ ~satx[[ix]] *), switchendHCLDIp[[provx[[ix]]]], 0], {ix, nx}];
ropminhosttandemHCx = Table[If[hostx[[ix]] ∧ ropx[[ix]] ∧ tandemx[[ix]] ≠ ix (* ∧ ~satx[[ix]] *),

```



```

switchendHCp[provx[ix]] - (locinHCp[provx[ix]] + locoutHCp[provx[ix]] - intraHCp[provx[ix]]), 0] +
If[satx[ix], intraHCp[provx[ix]], 0], {ix, nx}];
Print[TableForm[Table[{ix, y[ix, 13], ropminhosttandemHCLocx[ix], ropminhosttandemHCLDNx[ix],
ropminhosttandemHCLDix[ix], ropminhosttandemHCx[ix]}, {ix, nx}],
TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
TableHeadings → {None, {"ix", "Nombre Central", "ropminhosttandemHCLocx",
"ropminhosttandemHCLDNx", "ropminhosttandemHCLDix", "ropminhosttandemHCx"}}]];
(* Para las centrales que: son Host, son de provincias, no es satelital,
y no es considerada la Tandem de su departamento: Se toma el tráfico total considerado como conmutable a nivel
de toda su provincia restandole los siguiente: El tráfico total Intraprovincial Entrante de la provincia,
El tráfico total Intraprovincial Saliente de la provincia,
y el tráfico Intranodo. Nótese que el tráfico restado es el que no pasa por el nivel H-T *)

limaminhosttandemLocx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix (*^~satx[ix]*),
switchendLocp[provx[ix]] - (locinp[provx[ix]] + locoutp[provx[ix]] - intrap[provx[ix]]), 0] +
If[satx[ix], intrap[provx[ix]], 0], {ix, nx}];
limaminhosttandemLDNx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix
(*^~satx[ix]*), switchendLDNp[provx[ix]], 0], {ix, nx}];
limaminhosttandemLDix = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix
(*^~satx[ix]*), switchendLDip[provx[ix]], 0], {ix, nx}];
limaminhosttandemx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix (*^~satx[ix]*),
switchendp[provx[ix]] - (locinp[provx[ix]] + locoutp[provx[ix]] - intrap[provx[ix]]), 0] +
If[satx[ix], intrap[provx[ix]], 0], {ix, nx}];
Print[TableForm[Table[{ix, y[ix, 13], limaminhosttandemLocx[ix], limaminhosttandemLDNx[ix],
limaminhosttandemLDix[ix], limaminhosttandemx[ix]}, {ix, nx}], TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central",
"limaminhosttandemLocx", "limaminhosttandemLDNx", "limaminhosttandemLDix", "limaminhosttandemx"}}]];

limaminhosttandemHCLocx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix (*^~satx[ix]*),
switchendHCLocp[provx[ix]] - (locinHCp[provx[ix]] + locoutHCp[provx[ix]] - intraHCp[provx[ix]]) +
If[satx[ix], intraHCp[provx[ix]], 0], 0], {ix, nx}];
limaminhosttandemHCLDNx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix
(*^~satx[ix]*), switchendHCLDNp[provx[ix]], 0], {ix, nx}];
limaminhosttandemHCLDix = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix
(*^~satx[ix]*), switchendHCLDip[provx[ix]], 0], {ix, nx}];
limaminhosttandemHCx = Table[If[hostx[ix] ∧ limax[ix] ∧ tandemx[ix] ≠ ix (*^~satx[ix]*),
switchendHCp[provx[ix]] - (locinHCp[provx[ix]] + locoutHCp[provx[ix]] - intraHCp[provx[ix]]), 0] +
If[satx[ix], intraHCp[provx[ix]], 0], {ix, nx}];
Print[TableForm[Table[{ix, y[ix, 13], limaminhosttandemHCLocx[ix], limaminhosttandemHCLDNx[ix],
limaminhosttandemHCLDix[ix], limaminhosttandemHCx[ix]}, {ix, nx}],
TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},
TableHeadings → {None, {"ix", "Nombre Central", "limaminhosttandemHCLocx",
"limaminhosttandemHCLDNx", "limaminhosttandemHCLDix", "limaminhosttandemHCx"}}]];
(* Para las centrales que: son Host, son de Lima, no es satelital, y no es considerada la Tandem
de su departamento: Se toma el tráfico total considerado como conmutable a nivel de toda su
provincia restandole los siguiente: El tráfico total Intraprovincial Entrante de la provincia,
El tráfico total Intraprovincial Saliente de la provincia,
y el tráfico Intranodo. Nótese que el tráfico restado es el que no pasa por el nivel H-T *)

minhosttandemLocx = rop1x * ropminhosttandemLocx + lima1x * limaminhosttandemLocx;
minhosttandemLDNx = rop1x * ropminhosttandemLDNx + lima1x * limaminhosttandemLDNx;
minhosttandemLDix = rop1x * ropminhosttandemLDix + lima1x * limaminhosttandemLDix;
minhosttandemx = rop1x * ropminhosttandemx + lima1x * limaminhosttandemx;
Print[TableForm[
Table[{ix, y[ix, 13], minhosttandemLocx[ix], minhosttandemLDNx[ix], minhosttandemLDix[ix], minhosttandemx[ix]},
{ix, nx}], TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None,
{"ix", "Nombre Central", "minhosttandemLocx", "minhosttandemLDNx", "minhosttandemLDix", "minhosttandemx"}}]];

minhosttandemLocd = zerod;
minhosttandemLDNd = zerod;
minhosttandemLDId = zerod;
minhosttandemd = zerod;
Do[minhosttandemLocd[deptx[ix]] = minhosttandemLocd[deptx[ix]] + minhosttandemLocx[ix];
minhosttandemLDNd[deptx[ix]] = minhosttandemLDNd[deptx[ix]] + minhosttandemLDNx[ix];
minhosttandemLDId[deptx[ix]] = minhosttandemLDId[deptx[ix]] + minhosttandemLDix[ix];
minhosttandemd[deptx[ix]] = minhosttandemd[deptx[ix]] + minhosttandemx[ix], {ix, nx}];
Print[TableForm[Table[{id, deptTXTd[id], minhosttandemLocd[id], minhosttandemLDNd[id],
minhosttandemLDId[id], minhosttandemd[id]}, {id, nd}], TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento",
"minhosttandemLocd", "minhosttandemLDNd", "minhosttandemLDId", "minhosttandemd"}}]];

minhosttandemHCLocx = rop1x * ropminhosttandemHCLocx + lima1x * limaminhosttandemHCLocx;
minhosttandemHCLDNx = rop1x * ropminhosttandemHCLDNx + lima1x * limaminhosttandemHCLDNx;
minhosttandemHCLDix = rop1x * ropminhosttandemHCLDix + lima1x * limaminhosttandemHCLDix;
minhosttandemHCx = rop1x * ropminhosttandemHCx + lima1x * limaminhosttandemHCx;
Print[TableForm[Table[{ix, y[ix, 13], minhosttandemHCLocx[ix], minhosttandemHCLDNx[ix],
minhosttandemHCLDix[ix], minhosttandemHCx[ix]}, {ix, nx}], TableDirections → {Column, Row},
TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central",
"minhosttandemHCLocx", "minhosttandemHCLDNx", "minhosttandemHCLDix", "minhosttandemHCx"}}]];

```

```
(*Sumando los dos resultados previos (Lima+Provincias) obtenemos el tráfico
total que es conmutable a nivel H-T*)

loadhostttandemLocx =  $\frac{\text{minhostttandemHCLocx}}{60}$ ;

loadhostttandemLDNx =  $\frac{\text{minhostttandemHCLDNx}}{60}$ ;

loadhostttandemLDIx =  $\frac{\text{minhostttandemHCLDIX}}{60}$ ;

loadhostttandemx =  $\frac{\text{minhostttandemHCx}}{60}$ ;

Print[TableForm[Table[{ix, y[[ix, 13]], loadhostttandemLocx[[ix]], loadhostttandemLDNx[[ix]],
    loadhostttandemLDIx[[ix]], loadhostttandemx[[ix]]}, {ix, nx}], TableDirections -> {Column, Row},
    TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings -> {None, {"ix", "Nombre Central",
        "loadhostttandemLocx", "loadhostttandemLDNx", "loadhostttandemLDIx", "loadhostttandemx"}}]];

(*Para convertir los minutos a Erlangs*)
(* Se divide el tráfico total considerado como conmutable a nivel H-
T entre 60 para identificar las cargas en Erlangs *)
```

ESTIMACION DE TRONCALES Y E1s DE VOZ

```
Print[Style["FÓRMULA DE ERLANG B EN GENERAL",
    Bold, 16, Underlined, White, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["Definición de la función CalcTrunk"];

(* ERLANGB ProbLoss[λ_, n_] :=  $\frac{\lambda^n}{n!} / \sum_{j=0}^n \frac{\lambda^j}{j!}$ ; *)

(* Formula para hallar el número de troncales, basado en la fórmula de Erlang B *)

CalcTrunk[λ_, p_] := Module[{n, S}, n = 0; S = 1; If[λ > 0 ∧ p < 1, While[ $\frac{1}{S} > p$ , n = n + 1; S =  $\frac{S * n}{\lambda} + 1$ ];, n = 0]; n];

(* Se define esta función para evitar estar calculando manualmente
las cargas con procedimientos que son iguales a lo largo de todo el código. *)
```

```
Print[Style["ESTIMACIÓN DEL NÚMERO DE TRONCALES Y DE E1s DE VOZ",
    Bold, 16, Underlined, White, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print[Style["A NIVEL SATELITAL", Bold, 16, Underlined, White, Background -> Blue]];

trunksatx = zerox;
e1satx = zerox;
Do[If[deptsatx[[ix]] ≠ MADREDEDIOS, trunksatx[[ix]] = CalcTrunk[loadsatx[[ix]], prob];], {ix, nx}];

Do[e1satx[[ix]] = Ceiling[ $\frac{\text{Ceiling}[\frac{\text{trunksatx}[[ix]]}{30}]}{\text{compresDCME}}$ ], {ix, nx}];

Print[TableForm[
    Table[{ix, y[[ix, 13]], trunksatx[[ix]], e1satx[[ix]]}, {ix, Select[Range[nx], trunksatx[[#]] > 0 ∨ e1satx[[#]] > 0 &]}],
    TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1},
    TableHeadings -> {None, {"ix", "Nombre Central", "trunksatx", "e1satx"}}]];

(* Se estima el número de troncales a nivel Satelital sobre la base de las
cargas en Erlangs estimadas en la sección anterior *)

(* A partir del número de troncales estimados se estima el núemro de E1s
correspondientes. Para ello se divide el número de troncales entre 30 *)

Print[Style["A NIVEL DE RUTAS DIRECTAS", Bold, 16, Underlined, White, Background -> Blue]];

trunkRDx = Table[{loadRDx[[ird, 1]], loadRDx[[ird, 2]], CalcTrunk[loadRDx[[ird, 3]], prob]}, {ird, nrd}];
Print[TableForm[Table[{ird, trunkRDx[[ird, 1]], y[[trunkRDx[[ird, 1]], 13]],
    trunkRDx[[ird, 2]], y[[trunkRDx[[ird, 2]], 13]], trunkRDx[[ird, 3]]}, {ird, nrd}],
    TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1}, TableHeadings ->
    {None, {"ird", "ix Origen", "Nombre Central Origen", "ix Destino", "Nombre Central Destino", "trunkRDx"}}]];

e1RDx = zerox;

Do[e1RDx[[trunkRDx[[ird, 1]]]] = e1RDx[[trunkRDx[[ird, 1]]]] + Ceiling[ $\frac{\text{trunkRDx}[[ird, 3]]}{30 \text{ fiberfill}}$ ];, {ird, nrd}];

Print[TableForm[Table[{ix, y[[ix, 13]], e1RDx[[ix]]}, {ix, Select[Range[nx], e1RDx[[#]] > 0 &]}],
    TableDirections -> {Column, Row}, TableAlignments -> Right, TableSpacing -> {0, 1},
    TableHeadings -> {None, {"ix", "Nombre Central", "e1RDx"}}]];

(* Se estima el número de troncales correspondiente al tráfico de las Rutas Directas. La variable
loadRDx equivalente al total de las cargas en Erlangs que entran a cada central a través de RDs. Recordar
que para cada central se identificó las centrales que le envían tráfico mediante las RD. Luego,
se tomó la variable interprovouthCx o interprovincial saliente en minutos de HC de dichas
centrales y se multiplicó por su respectivo % de tráfico saliente a través de las RD,
se dividió entre 60 para tener Erlangs. Luego, para cada una de las centrales de destino se realizó
la sumatoria de los resultados estimados en las centrales que le envían tráfico por las RD *)

(* A partir del número de troncales de las RDS se estima el núemro de E1s
correspondientes. Se divide entre 30 y entre el factor de utilización *)
```

"A NIVEL REMOTA-CABECERAS", Bold, 16, Underlined, White, Background → Blue];

trunkremotehostx = zerox;

e1remotehostx = zerox;

Do[trunkremotehostx[[ix]] = CalcTrunk[loadremotehostx[[ix]], prob];

If[fiberx[[ix]], e1remotehostx[[ix]] = Ceiling[$\frac{\text{trunkremotehostx}[[ix]]}{30 \cdot \text{fiberfill}}$];,

If[satx[[ix]], e1remotehostx[[ix]] = Ceiling[$\frac{\text{trunkremotehostx}[[ix]]}{30}$];,

e1remotehostx[[ix]] = Ceiling[$\frac{\text{trunkremotehostx}[[ix]]}{30 \cdot \text{radiofill}}$];];];, {ix, nx}];

Print[TableForm[Table[{ix, y[[ix, 13]], trunkremotehostx[[ix]], e1remotehostx[[ix]]},

{ix, Select[Range[nx], trunkremotehostx[[#]] > 0 ∨ e1remotehostx[[#]] > 0 &]}],

TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},

TableHeadings → {None, {"ix", "Nombre Central", "trunkremotehostx", "e1remotehostx"}}]]];

(* Se estima el número de troncales a nivel RH sobre la base de las cargas en Erlangs estimadas en la sección anterior *)

(* A partir del número de troncales estimados en el tramo RH se estima el número de E1s correspondientes. Se divide entre 30 y entre el factor de utilización que corresponde según la tecnología de transmisión empleada en cada central *)

TOTALE1 = 0;

TOTALE1 = Plus @@ e1remotehostx;

"A NIVEL CABECERAS-TANDEM", Bold, 16, Underlined, White, Background → Blue];

trunkhosttandemx = zerox;

e1hosttandemx = zerox;

Do[If[host1x[[ix]] == 1 ∧ tandemx[[ix]] == idinvertx[[248]],

trunkhosttandemx[[ix]] = CalcTrunk[$\frac{\text{loadhosttandemx}[[ix]]}{2}$, prob];,

trunkhosttandemx[[ix]] = CalcTrunk[loadhosttandemx[[ix]], prob];];, {ix, nx}];

Do[If[fiberx[[ix]], e1hosttandemx[[ix]] = Ceiling[$\frac{\text{trunkhosttandemx}[[ix]]}{30 \cdot \text{fiberfill}}$];,

If[satx[[ix]], e1hosttandemx[[ix]] = Ceiling[$\frac{\text{trunkhosttandemx}[[ix]]}{30 \cdot \text{compresDCME}}$];,

e1hosttandemx[[ix]] = Ceiling[$\frac{\text{trunkhosttandemx}[[ix]]}{30 \cdot \text{radiofill}}$];];];, {ix, nx}];

Do[If[host1x[[ix]] == 1 ∧ tandemx[[ix]] == idinvertx[[248]],

If[sat1x[[ix]] == 1, e1hosttandemx[[ix]] = Ceiling[$\frac{2 \cdot \text{e1hosttandemx}[[ix]]}{\text{compresDCME}}$];,

e1hosttandemx[[ix]] = 2 e1hosttandemx[[ix]];];];, {ix, nx}];

e1hosttandemx = e1hosttandemx + e1RDx;

Print[TableForm[Table[{ix, y[[ix, 13]], trunkhosttandemx[[ix]], e1hosttandemx[[ix]]},

{ix, Select[Range[nx], trunkhosttandemx[[#]] > 0 ∨ e1hosttandemx[[#]] > 0 &]}],

TableDirections → {Column, Row}, TableAlignments → Right, TableSpacing → {0, 1},

TableHeadings → {None, {"ix", "Nombre Central", "trunkhosttandemx", "e1hosttandemx"}}]]];

(* Se estima el número de troncales a nivel HT sobre la base de las cargas en Erlangs estimadas en la sección anterior *)

(* A partir del número de troncales estimados en el tramo HT se estima el número de E1s correspondientes. Se divide entre 30 y entre el factor de utilización que corresponde según la tecnología de transmisión empleada en cada central *)

TOTALE1 = TOTALE1 + Plus @@ e1remotehostx;

trunkrdx = zerox;

Do[trunkrdx[[trunkRDx[[ird, 1]]]] = trunkrdx[[trunkRDx[[ird, 1]]]] + trunkRDx[[ird, 3]];

trunkrdx[[trunkRDx[[ird, 2]]]] = trunkrdx[[trunkRDx[[ird, 2]]]] + trunkRDx[[ird, 3]];];, {ird, nrd}];

trunkx = Table[remote1x[[ix]] * trunkremotehostx[[ix]] +

host1x[[ix]] * (trunkhosttandemx[[ix]] * If[deptsx[[ix]] == LIMA, 2, 1] + trunkrdx[[ix]]), {ix, nx}];

Do[If[remotex[[ix]], trunkx[[allhostx[[ix]]]] = trunkx[[allhostx[[ix]]]] + trunkremotehostx[[ix]]];

If[hostx[[ix]] ∧ tandemx[[ix]] ≠ ix, trunkx[[tandemx[[ix]]]] = trunkx[[tandemx[[ix]]]] + trunkhosttandemx[[ix]];];, {ix, nx}];

Print[TableForm[Table[{ix, y[[ix, 13]], trunkx[[ix]]}, {ix, nx}], TableDirections → {Column, Row},

TableAlignments → Right, TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "Nombre Central", "trunkx"}}]]];

(* Identificación del total de troncales asociadas a cada central. Para las centrales remotas es aplicable el total trunkremotehostx, mientras que para las cabeceras es aplicable el total trunkhosttandemx + trunksatx. A las centrales cabeceras les agrega las troncales de sus remotas asociadas (los trunkremotehostx de sus remotas). A las centrales Tandem le suma además las troncales de sus cabeceras asociadas (los trunkhosttandemx de sus cabeceras asociadas *)

"ESTIMACIÓN DE CARGAS EN E1s ASOCIADAS AL SERVICIO ADSL", Bold, 16, Underlined, White, Background → RGBColor[0, $\frac{1}{2}$, 0]]];

"CIRCUITOS ADSL", Bold, 16, Underlined, White, Background → Blue];


```

Print["demandasADSL1"];
OpenRead["Demandas_ADSL_2005OrdTotal_6Pops.txt"];
demandasADSL1 = ReadList["Demandas_ADSL_2005OrdTotal_6Pops.txt",
  {Word, Word, Word, Real, Real, Real, Real, Real, Real, Real}];
Close["Demandas_ADSL_2005OrdTotal_6Pops.txt"];

Print["Localidades"];
Localidades = Length[demandasADSL1];
Print[Localidades];
(* Todas las Localidades con servicio ADSL *)
Print[TableForm[Table[Prepend[Table[demandasADSL1[[i, j]], {j, 1, 10}], i], {i, Localidades}],
  TableDirections -> {Column, Row}, TableAlignments -> Right,
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"Nro", "Departamento", "Nombre",
    "Pop", "Id Central", "2000", "600", "400", "200", "100", "Plus"}}]];
(*Importación de la demanda de cada año*)

Print["equivADSL"];
equivADSL = Table[{i, idinvertx[[demandasADSL1[[i, 4]]]], {i, Localidades}}];
Print[TableFormequivADSL];
(*Se utiliza un input para la equivalencia entre Localidad ADSL y
  Central en la red Fija Ya no se utiliza el archivo input_equiv_ADSL.txt*)

Print["neq"];
neq = Length[equivADSL];
Print[neq];

Print["EquivADSLFija"];
EquivADSLFija = Table[0, {ix, nx}];
Do[
  If[equivADSL[[ieq, 2]] != 0, EquivADSLFija[[equivADSL[[ieq, 2]]]] = equivADSL[[ieq, 1]]];, {ieq, neq}];
Print[EquivADSLFija];

(* lista de nodos ATM en Lima*)
OpenRead["NodosATMLima_6Pops.txt"];
(*Instruccion escondida. En texto blanco*)
CxATMLima1x = ReadList["NodosATMLima_6Pops.txt", Real];
CxATMLimax = zerox;
Do[CxATMLimax[[idinvertx[[ix]]]] = CxATMLima1x[[ix]], {ix, nx}];
Print[CxATMLimax];

Print["sumademandasADSL"];
sumademandasADSL =
  Table[demandasADSL1[[i, 5]] + demandasADSL1[[i, 6]] + demandasADSL1[[i, 7]] + demandasADSL1[[i, 8]] +
    demandasADSL1[[i, 9]] + demandasADSL1[[i, 10]], {i, Localidades}]; Print[sumademandasADSL];
(*Suma las demandas *)

Print["demandasADSL1x"];
demandasADSL1x = Table[Table[0, {il, 6}], {ix, nx}];
Do[If[EquivADSLFija[[ix]] != 0,
  auxLoc = EquivADSLFija[[ix]];
  demandasADSL1x[[ix, 1]] = demandasADSL1x[[ix, 1]] + demandasADSL1[[auxLoc, 5]];
  demandasADSL1x[[ix, 2]] = demandasADSL1x[[ix, 2]] + demandasADSL1[[auxLoc, 6]];
  demandasADSL1x[[ix, 3]] = demandasADSL1x[[ix, 3]] + demandasADSL1[[auxLoc, 7]];
  demandasADSL1x[[ix, 4]] = demandasADSL1x[[ix, 4]] + demandasADSL1[[auxLoc, 8]];
  demandasADSL1x[[ix, 5]] = demandasADSL1x[[ix, 5]] + demandasADSL1[[auxLoc, 9]];
  demandasADSL1x[[ix, 6]] = demandasADSL1x[[ix, 6]] + demandasADSL1[[auxLoc, 10]]];, {ix, nx}];
Print[TableForm[demandasADSL1x]];
(*Distribución de las demandas a nivel departamental*)

Print["demandasADSL1Nuc"];
demandasADSL1Nuc = demandasADSL1x;
Print[TableForm[demandasADSL1Nuc]];
(*Distribución de las demandas para Nuc*)

Print["sumademandasADSLNuc"];
sumademandasADSLNuc =
  Table[demandasADSL1Nuc[[i, 1]] + demandasADSL1Nuc[[i, 2]] + demandasADSL1Nuc[[i, 3]] + demandasADSL1Nuc[[
    i, 4]] + demandasADSL1Nuc[[i, 5]] + demandasADSL1Nuc[[i, 6]], {i, nx}]; Print[sumademandasADSLNuc];
(*Suma las demandas Nuc*)

```

```

Print["SumDemADSL"];
SumDemADSL = Plus @@ sumademandasADSL;
Print[SumDemADSL];
(*Suma las demandas Total *)

Print["demandaADSL2000"];
demandaADSL2000 = Table[demandasADSL1[[i, 5]], {i, Localidades}];
Print[demandaADSL2000];

Print["demandaADSL2000Nuc"];
demandaADSL2000Nuc = Table[demandasADSL1Nuc[[ix, 1]], {ix, nx}];
Print[demandaADSL2000Nuc];

Print["SumComprobdemandaADSL2000"];
SumComprobdemandaADSL2000 = Plus @@ demandaADSL2000;
Print[SumComprobdemandaADSL2000];

Print["CentralesConExtensores"];
CentralesConExtensores = Ceiling[ $\frac{\text{demandaADSL2000}}{40}$ ];
(*CentralesConExtensores=Ceiling[ $\frac{\text{sumademandasADSL}}{40}$ ];*)
Print[CentralesConExtensores];
(*Se ha modificado la expresion de tal manera que unicamente
se consideren las lineas de 2Mbps para el empleo de los extensores *)
(* Uso de extensores. *)
Print["SumCentralesConExtensores"];
SumCentralesConExtensores = Plus @@ CentralesConExtensores;
Print[SumCentralesConExtensores];
(*Se a adio instruccion de suma de comprobacion de
las lineas que requieres adecuacion utilizando extensores*)

Print[Style["CARGA DE VELOCIDAD", Bold, 16, Underlined, White, Background -> Blue]];

OpenRead["Servicios_ADSLCambios.txt"];
velocidadADSL = ReadList["Servicios_ADSLCambios.txt", {Word, Real}];
Close["Servicios_ADSLCambios.txt"];
(*Carga las velocidades de cada uno de los tipos de servicios de ADSL*)

Print["velocidad2000"];
velocidad2000 = velocidadADSL[[1]][2];
Print[velocidad2000];
(*Carga las velocidad 2000 en fraccion de Mbps*)

Print["velocidad1200"];
velocidad1200 = velocidadADSL[[2]][2];
Print[velocidad1200];
(*Se ha cambiado 600 por 1200 para que corresponda con
el valor que esta leyendo del archivo Servicios_ADSLCambios.txt*)
(*velocidad600=velocidadADSL[[2]][2];*)

Print["velocidad900"];
velocidad900 = velocidadADSL[[3]][2];
Print[velocidad900];
(*Se ha cambiado 400 por 900 para que corresponda con el
valor que esta leyendo del archivo Servicios_ADSLCambios.txt*)
(*velocidad400=velocidadADSL[[3]][2];*)

Print["velocidad600"];
velocidad600 = velocidadADSL[[4]][2];
Print[velocidad600];
(*Se ha cambiado 200 por 600 para que corresponda con el
valor que esta leyendo del archivo Servicios_ADSLCambios.txt*)
(*velocidad200=velocidadADSL[[4]][2];*)

Print["velocidad400"];
velocidad400 = velocidadADSL[[5]][2];
Print[velocidad400];
(*Se ha cambiado 100 por 400 para que corresponda con el

```

```

    valor que esta leyendo del archivo Servicios_ADSLCambios.txt*)
(*velocidad100=velocidadADSL[[5]][2];*)

Print["velocidadPlus"];
velocidadPlus = velocidadADSL[[6]][2];
Print[velocidadPlus];
(*Carga las velocidad Plus en fraccion de Mbps*)

Print["porcentajeBWGarant"];
porcentajeBWGarant = velocidadADSL[[7]][2];
Print[porcentajeBWGarant];
(*Carga las velocidad Plus en fraccion de Mbps*)

Print["porcentajeBWGarantPlus"];
porcentajeBWGarantPlus = porcentajeBWGarant;
Print[porcentajeBWGarantPlus];
(*Carga las velocidad Plus en fraccion de Mbps*)

(*Calcula el Ancho de Banda de seguridad*)
(*Observacion: Como se variaron los nombres de las velocidades
en el paso anterior se ha cambiado la siguiente instruccion para que
pueda leer los valores del archivo de entrada Servicios_ADSLCambios*)
(*BWGarant=Table[(demandasADSL[[a,i,4]]*velocidad2000+demandasADSL[[a,i,5]]*velocidad600+
demandasADSL[[a,i,6]]*velocidad400+demandasADSL[[a,i,7]]*velocidad200+
demandasADSL[[a,i,8]]*velocidad100)*porcentajeBWGarant+
demandasADSL[[a,i,9]]*velocidadPlus*porcentajeBWGarantPlus),{a,Años},{i,Localidades}];*)
(*Se ha añadido 10% de Overbooking (cómo parámetro factrednd) en el calculo
del ancho de banda de seguridad para aprovechar las características
estadística de las transmisiones en rafagas de los clientes,
aumentando así la utilización de los recursos de red*)
Print["BWGarant"];
BWGarant = Table[(demandasADSL1[[i, 5]] * velocidad2000 + demandasADSL1[[i, 6]] * velocidad1200 +
demandasADSL1[[i, 7]] * velocidad900 + demandasADSL1[[i, 8]] * velocidad600 +
demandasADSL1[[i, 9]] * velocidad400) * porcentajeBWGarant + demandasADSL1[[i, 10]] *
velocidadPlus * porcentajeBWGarantPlus), {i, Localidades}] * factrednd;
Print[BWGarant]; BWGarantNuc = Table[(demandasADSL1Nuc[[in, 1]] * velocidad2000 +
demandasADSL1Nuc[[in, 2]] * velocidad1200 + demandasADSL1Nuc[[in, 3]] * velocidad900 +
demandasADSL1Nuc[[in, 4]] * velocidad600 + demandasADSL1Nuc[[in, 5]] * velocidad400) *
porcentajeBWGarant + demandasADSL1Nuc[[in, 6]] * velocidadPlus *
porcentajeBWGarantPlus), {in, nx}] * factrednd;
Print[BWGarantNuc];
(* Calcula el Ancho de Banda de seguridad,
usamos aquí y sólo aquí el factor de redundancia *)

Print["SumBWGarant"];
SumBWGarant = Plus @@ BWGarant;
Print[SumBWGarant];

Print["SumBWGarantNuc"];
SumBWGarantNuc = Plus @@ BWGarantNuc;
Print[SumBWGarantNuc];

Print[
Style["PRECIOS DE CLUSTER_FULL, SHELF_FULL, TARJ_LT_24, SHELF_ARMARIO,
SHELF_EXTENSION, TARJ_NT_8_MBPS, TARJ_NT_34_MBPS,
TARJ_NT_155_MBPS", Bold, 16, Underlined, White, Background -> Blue],
];

Print["preciosADSL"];
OpenRead["Precios_DSLAM.txt"];
preciosADSL = ReadList["Precios_DSLAM.txt", {Word, Real}];
Close["Precios_DSLAM.txt"];
Print[preciosADSL];

Print["precioClusterFull"];
precioClusterFull = preciosADSL[[1]][2];
Print[precioClusterFull];

Print["precioShelfFull"];

```

```
precioShelfFull = preciosADSL[[2]][2];
Print[precioShelfFull];

Print["precioTarjLT24"];
precioTarjLT24 = preciosADSL[[3]][2];
Print[precioTarjLT24];

Print["precioShelf"];
precioShelf = preciosADSL[[4]][2];
Print[precioShelf];

Print["precioArmario"];
precioArmario = preciosADSL[[5]][2];
Print[precioArmario];

Print["precioShelfExtension"];
precioShelfExtension = preciosADSL[[6]][2];
Print[precioShelfExtension];

Print["precioTarjNT8"];
precioTarjNT8 = preciosADSL[[7]][2];
Print[precioTarjNT8];

Print["precioTarjNT34"];
precioTarjNT34 = preciosADSL[[8]][2];
Print[precioTarjNT34];

Print["precioTarjNT155"];
precioTarjNT155 = preciosADSL[[9]][2];
Print[precioTarjNT155];

Print["precioPreconcentrClusterFull"];
precioPreconcentrClusterFull = preciosADSL[[10]][2];
Print[precioPreconcentrClusterFull];

Print["precioPreconcentrShelfFull"];
precioPreconcentrShelfFull = preciosADSL[[11]][2];
Print[precioPreconcentrShelfFull];

Print["precioTarjLT4E1"];
precioTarjLT4E1 = preciosADSL[[12]][2];
Print[precioTarjLT4E1];

Print["precioExtensorADSL"];
precioExtensorADSL = preciosADSL[[13]][2];
Print[precioExtensorADSL];

Print[Style[
  "Parámetros de Máximo número de usuarios por componente: Cluster, Shelf y Tarj", Bold, 16, Underlined, White, Background -> Blue],];

Print["parametrosDSLAM"];
OpenRead["Parametros_DSLAM.txt"];
parametrosDSLAM = ReadList["Parametros_DSLAM.txt", {Word, Real}];
Close["Parametros_DSLAM.txt"];
Print[parametrosDSLAM];
(*Parámetros de Máximo número de usuarios por componente: Cluster, Shelf y Tarj*)

Print["maxUsrCluster"];
maxUsrCluster = parametrosDSLAM[[1]][2];
Print[maxUsrCluster];
(* Parámetros de Máximo número de usuarios por Cluster *)

Print["maxUsrShelf"];
maxUsrShelf = parametrosDSLAM[[2]][2];
Print[maxUsrShelf];
(* Parámetros de Máximo número de usuarios por Shelf *)

Print["maxUsrTarj"];
```

```

maxUsrTarj = parametrosDSLAM[[3]][2];
Print[maxUsrTarj];
(* Parámetros de Máximo número de usuarios por Tarjeta *)

Print[Style["Número de componentes que forman los DSLAMs por ubicación",
  Bold, 16, Underlined, White, Background -> Blue]];

Print["numeroClusterFull"];
numeroClusterFull = Table[Floor[ $\frac{\text{sumademandasADSL}[[i]]}{\text{maxUsrCluster}}$ ], {i, Localidades}];
Print[numeroClusterFull];
(* Número de los Clusters completos *)

Print["numeroClusterFullNuc"];
numeroClusterFullNuc = Floor[ $\frac{\text{sumademandasADSLNuc}}{\text{maxUsrCluster}}$ ];
Print[numeroClusterFullNuc];
(* Número de los Clusters completos Nuc *)

Print["SumnumeroClusterFull"];
Sum numeroClusterFull = Plus @@ numeroClusterFull;
Print[SumnumeroClusterFull];
(*Instruccion añadida para verificacion de suma de numero de clusters totalmente ocupados*)

Print["numeroShelfFull"];
numeroShelfFull =
  Table[Floor[ $\frac{\text{sumademandasADSL}[[i]] - \text{numeroClusterFull}[[i]] * \text{maxUsrCluster}}{\text{maxUsrShelf}}$ ], {i, Localidades}];
Print[numeroShelfFull];
(* Número de los Armarios completos *)

Print["numeroShelfFullNuc"];
numeroShelfFullNuc = Floor[ $\frac{\text{sumademandasADSLNuc} - \text{numeroClusterFullNuc} * \text{maxUsrCluster}}{\text{maxUsrShelf}}$ ];
Print[numeroShelfFullNuc];
(* Número de los Armarios completos Nuc *)

Print["SumnumeroShelfFull"];
Sum numeroShelfFull = Plus @@ numeroShelfFull;
Print[Sum numeroShelfFull];
(*Instruccion añadida para verificacion de suma de numero de Shelves completamente ocupados*)

Print["numeroTarjLT24"];
numeroTarjLT24 = Table[Ceiling[ $\frac{\text{sumademandasADSL}[[i]] - \text{numeroClusterFull}[[i]] * \text{maxUsrCluster} - \text{numeroShelfFull}[[i]] * \text{maxUsrShelf}}{\text{maxUsrTarj}}$ ],
  {i, Localidades}];
Print[numeroTarjLT24];
(* Número de Tarjetas de Terminación de Línea *)

Print["nuermoTarjLT24Nuc"];
numeroTarjLT24Nuc = Ceiling[ $\frac{\text{sumademandasADSLNuc} - \text{numeroClusterFullNuc} * \text{maxUsrCluster} - \text{numeroShelfFullNuc} * \text{maxUsrShelf}}{\text{maxUsrTarj}}$ ];
Print[numeroTarjLT24Nuc];
(* Número de Tarjetas de Terminación de Línea Nuc*)

Print["SumnumeroTarjLT24"];
SumnumeroTarjLT24 = Plus @@ numeroTarjLT24;
Print[Sum numeroTarjLT24];
(*Instruccion añadida para verificacion de
suma de numero de Tarjetas de Terminacion de Linea*)

Print["numeroArmario"];
numeroArmario =
  Table[Ceiling[ $\frac{\text{sumademandasADSL}[[i]] - \text{numeroClusterFull}[[i]] * \text{maxUsrCluster}}{2 \text{ maxUsrShelf}}$ ], {i, Localidades}];

```



```

Print[numeroArmario];

Print["numeroArmarioNuc"];
numeroArmarioNuc = Ceiling[ $\frac{\text{sumademandasADSLNuc} - \text{numeroClusterFullNuc} * \text{maxUsrCluster}}{2 \text{ maxUsrShelf}}$ ];
Print[numeroArmarioNuc];

Print["SumnumeroArmario"];
SumnumeroArmario = Plus@@numeroArmario;
Print[SumnumeroArmario];
(*Instruccion a adida para verificacion de suma de numero de Armarios*)

Print["CostoDSLAMSintarjetaNT"];
CostoDSLAMSintarjetaNT =
  Table[(numeroClusterFull[[i]] * precioClusterFull + numeroShelfFull[[i]] * precioShelfFull +
    numeroTarjLT24[[i]] * precioTarjLT24 + numeroArmario[[i]] * precioArmario +
    (If[numeroShelfFull[[i]] > 1, If[numeroTarjLT24[[i]] > 0, numeroShelfFull[[i]],
      (numeroShelfFull[[i]] - 1)], 0)) * precioShelfExtension +
    (If[numeroTarjLT24[[i]] > 0, precioShelf, 0])), {i, Localidades}];
Print[CostoDSLAMSintarjetaNT];

Print["CostoDSLAMSintarjetaNTNuc"];
CostoDSLAMSintarjetaNTNuc = Table[0, {i, nx}];
Print[CostoDSLAMSintarjetaNTNuc];

Print["SumCostoDSLAMSintarjetaNT"];
SumCostoDSLAMSintarjetaNT = Plus@@CostoDSLAMSintarjetaNT;
Print[SumCostoDSLAMSintarjetaNT];
(*Instruccion a adida para verificacion de suma
del costo de DSLAM sin Tarjeta de Terminacion de Red*)

Print["BWClusterFull"];
BWClusterFull = Table[If[CostoDSLAMSintarjetaNT[[i]] == 0, 0,
  BWGarant[[i]] (numeroClusterFull[[i]] maxUsrCluster) / (numeroClusterFull[[i]] maxUsrCluster +
    numeroShelfFull[[i]] maxUsrShelf + numeroTarjLT24[[i]] maxUsrTarj)], {i, Localidades}];
Print[BWClusterFull];

Print["BWClusterFullNuc"];
BWClusterFullNuc = Table[BWGarantNuc[[i]], {i, nx}];
Print[BWClusterFullNuc];

Print["SumBWClusterFull"];
SumBWClusterFull = Plus@@BWClusterFull;
Print[SumBWClusterFull];
(*Instruccion a adida para verificacion de suma
de los anchos de banda de Clusters Totalmente ocupados*)

Print["BWClusterParcial"];
BWClusterParcial = Table[If[CostoDSLAMSintarjetaNT[[i]] == 0, 0,
  BWGarant[[i]] (numeroShelfFull[[i]] * maxUsrShelf + numeroTarjLT24[[i]] * maxUsrTarj) /
    (numeroClusterFull[[i]] maxUsrCluster + numeroShelfFull[[i]] maxUsrShelf +
    numeroTarjLT24[[i]] maxUsrTarj)], {i, Localidades}];
Print[BWClusterParcial];
(*Calcula el ancho de banda del Cluster Parcialmente ocupado*)

Print["BWClusterParcialNuc"];
BWClusterParcialNuc = Table[0, {i, nx}];
Print[BWClusterParcialNuc];
(*Calcula el ancho de banda del Cluster Parcialmente ocupado Nuc*)

Print["SumBWClusterParcial"];
SumBWClusterParcial = Plus@@BWClusterParcial;
Print[SumBWClusterParcial];
(*Instruccion a adida para verificacion de suma
de los anchos de banda de Clusters parcialmente ocupados*)

Print["TarjNTClusterFull"];
TarjNTClusterFull =

```



```

Table[If[BWClusterFull[[i]] > 0, If[BWClusterFull[[i]] ≤ 8, 8, If[ $\frac{\text{BWClusterFull}[[i]]}{\text{numeroClusterFull}[[i]]} \leq 34,$ 
34, If[ $\frac{\text{BWClusterFull}[[i]]}{\text{numeroClusterFull}[[i]]} \leq 155, 155, \text{"No"}]]], 0], \{i, \text{Localidades}\}];
Print[TarjNTClusterFull];

Print["TarjNTClusterFullNuc"];
TarjNTClusterFullNuc = Table[0, \{i, nx\}];
Print[TarjNTClusterFullNuc];

Print["TarjNTClusterParcial"];
TarjNTClusterParcial =
Table[(If[BWClusterParcial[[i]] > 0, If[BWClusterParcial[[i]] ≤ 8, 8, If[BWClusterParcial[[i]] ≤ 34,
34, If[BWClusterParcial[[i]] ≤ 155, 155, "No"]]], 0)], \{i, Localidades\}];
Print[TarjNTClusterParcial];

Print["TarjNTClusterParcialNuc"];
TarjNTClusterParcialNuc = Table[0, \{i, nx\}];
Print[TarjNTClusterParcialNuc];

Print["SumTarjNTClusterParcial"];
SumTarjNTClusterParcial = Plus @@ TarjNTClusterParcial;
Print[SumTarjNTClusterParcial];

(*Instruccion añadida para verificacion de suma de las velocidades de las Tarjetas de
Terminacion para Clusters parcialmente ocupados*)Print["CostoDSLAMConTarjetaNT1"];
CostoDSLAMConTarjetaNT1 = Table[(CostoDSLAMsintarjetaNT[[i]] +
(If[TarjNTClusterFull[[i]] == 8, precioTarjNT8, If[TarjNTClusterFull[[i]] == 34,
precioTarjNT34, If[TarjNTClusterFull[[i]] == 155, precioTarjNT155, 0]])) *
numeroClusterFull[[i]] + (If[TarjNTClusterParcial[[i]] == 8, precioTarjNT8,
If[TarjNTClusterParcial[[i]] == 34, precioTarjNT34,
If[TarjNTClusterParcial[[i]] == 155, precioTarjNT155, 0]]))), \{i, Localidades\}];
Plus @@ CostoDSLAMConTarjetaNT1[[1]];
Print[CostoDSLAMConTarjetaNT1];

(*Calcula el costo del DSLAM con Tarjeta de Terminacion de Red*)

Print["CostoDSLAMConTarjetaNT1"];
CostoDSLAMConTarjetaNT1 = Table[(CostoDSLAMsintarjetaNT[[i]] +
(If[TarjNTClusterFull[[i]] == 8, precioTarjNT8, If[TarjNTClusterFull[[i]] == 34,
precioTarjNT34, If[TarjNTClusterFull[[i]] == 155, precioTarjNT155, 0]])) *
numeroClusterFull[[i]] + (If[TarjNTClusterParcial[[i]] == 8, precioTarjNT8,
If[TarjNTClusterParcial[[i]] == 34, precioTarjNT34,
If[TarjNTClusterParcial[[i]] == 155, precioTarjNT155, 0]]))), \{i, Localidades\}];
Plus @@ CostoDSLAMConTarjetaNT1[[1]];
Print[CostoDSLAMConTarjetaNT1];

(*Calcula el costo del DSLAM con Tarjeta de Terminacion de Red*)
Print["SumCostoDSLAMConTarjetaNT1"];
SumCostoDSLAMConTarjetaNT1 = Plus @@ CostoDSLAMConTarjetaNT1;
Print[SumCostoDSLAMConTarjetaNT1];

(*Instruccion añadida para verificacion de suma
de los costos del DSLAM con Tarjeta de Terminacion de Red*)
Print["CostoDSLAMConTarjetaNT"];
CostoDSLAMConTarjetaNT = CostoDSLAMConTarjetaNT1 + (precioExtensorADSL CentralesConExtensores);
Print[CostoDSLAMConTarjetaNT];

(* Coste de los extensores, en caso de que sean necesarios*)
(*Calcula el costo del DSLAM con Tarjeta de Terminacion de Red mas el costo del extensor
de bucle de abonado donde sean necesarios*)Print["SumCostoDSLAMConTarjetaNT"];
SumCostoDSLAMConTarjetaNT = Plus @@ CostoDSLAMConTarjetaNT;
Print[SumCostoDSLAMConTarjetaNT];

(*Instruccion añadida para verificacion de suma de los costos del DSLAM con Tarjeta
de Terminacion de Red mas los costos de los extensores de bucle de abonado*)

Print["InterfazSTM1HaciaPOP"];
InterfazSTM1HaciaPOP = Table[If[TarjNTClusterFull[[i]] == 155, numeroClusterFull[[i]], 0] +
If[TarjNTClusterParcial[[i]] == 155, 1, 0], \{i, Localidades\}];
Print[InterfazSTM1HaciaPOP];

Print["InterfazSTM1HaciaPOPNUC"];
InterfazSTM1HaciaPOPNUC = Table[0, \{i, nx\}];
Print[InterfazSTM1HaciaPOPNUC];$ 
```

```

Print["SumInterfazSTM1HaciaPOP"];
SumInterfazSTM1HaciaPOP = Plus @@ InterfazSTM1HaciaPOP;
Print[SumInterfazSTM1HaciaPOP];
(*Instruccion añadida para verificacion de suma de las interfaces STM-1 hacia los PoPs ATM*)

Print["InterfazE3HaciaPOP"];
InterfazE3HaciaPOP = Table[If[TarjNTClusterFull[[i]] == 34, numeroClusterFull[[i]], 0] +
  If[TarjNTClusterParcial[[i]] == 34, 1, 0], {i, Localidades}];
Print[InterfazE3HaciaPOP];

Print["InterfazE3HaciaPOPNUC"];
InterfazE3HaciaPOPNUC = Table[0, {i, nx}];
Print[InterfazE3HaciaPOPNUC];

Print["SumInterfazE3HaciaPOP"];
SumInterfazE3HaciaPOP = Plus @@ InterfazE3HaciaPOP;
Print[SumInterfazE3HaciaPOP];
(*Instruccion añadida para verificacion de suma de las interfaces E-3 hacia los PoPs ATM*)
Print["InterfazE1HaciaPOP"];
InterfazE1HaciaPOP =
  Table[If[TarjNTClusterParcial[[i]] == 8, Ceiling[ $\frac{\text{BWClusterParcial}[[i]]}{2.048}$ ], 0], {i, Localidades}];
Print[InterfazE1HaciaPOP];

Print["InterfazE1HaciaPOPNUC"];
InterfazE1HaciaPOPNUC = Table[ $\frac{\text{BWClusterFullNuc}[[i]]}{2.048}$ , {i, nx}];
Print[InterfazE1HaciaPOPNUC];

Print["SumInterfazE1HaciaPOP"];
SumInterfazE1HaciaPOP = Plus @@ InterfazE1HaciaPOP;
Print[SumInterfazE1HaciaPOP];
(*Instruccion añadida para verificacion de suma de las interfaces E-1 hacia los PoPs ATM*)

Print["NumE1Equivs"];
NumE1Equivs = Table[63 * InterfazSTM1HaciaPOP[[i]] +
  16 * InterfazE3HaciaPOP[[i]] + InterfazE1HaciaPOP[[i]], {i, Localidades}];
Print[NumE1Equivs];
(*Calcula el numero de E1s equivalentes por site hacia los PoPs ATM*)

Print["NumE1EquivsNuc"];
NumE1EquivsNuc = Table[InterfazE1HaciaPOPNUC[[i]], {i, nx}];
Print[NumE1EquivsNuc];
(*Calcula el numero de E1s equivalentes por site hacia los PoPs ATM*)

Print["SumNumE1Equivs"];
SumNumE1Equivs = Plus @@ NumE1Equivs;
Print[SumNumE1Equivs];
(*Instruccion añadida para verificacion de suma del numero de E-
1 equivalentes hacia los PoPs ATM*) (*Carga los datos de potencia y energía*)

OpenRead["Energia_DSLAM.txt"];
energiaADSL = ReadList["Energia_DSLAM.txt", {Word, Real}];
Close["Energia_DSLAM.txt"];

Print["CostePorWDSLAM"];
CostePorWDSLAM = energiaADSL[[1]][2];
Print[CostePorWDSLAM];

Print["WPorLineaDSLAM"];
WPorLineaDSLAM = energiaADSL[[2]][2];
Print[WPorLineaDSLAM];

(*Calcula el costo del consumo de energia del DSLAM por watio por site*)
Print["CostoEnergiaDSLAM"];
CostoEnergiaDSLAM =
  Table[CostePorWDSLAM * WPorLineaDSLAM * sumademandasADSL[[i]], {i, Localidades}];
Print[CostoEnergiaDSLAM];

```

```
(*Instruccion añadida para verificacion de suma
de los costos de energia de los DSLAM por watio por site*)
Print["SumCostoEnergiaADSLAM"]; SumCostoEnergiaADSLAM = Plus @@ CostoEnergiaADSLAM;
Print[SumCostoEnergiaADSLAM];

Print[Style["REPARTO A SERVICIOS POR DEPARTAMENTO",
  Bold, 16, Underlined, White, Background -> Blue]];

(*Carga de los ficheros de entrada*)
OpenRead["PuertosATM155ParaGigADSL_LIMA6PE3T0T.txt"];
PuertosLima =
  ReadList["PuertosATM155ParaGigADSL_LIMA6PE3T0T.txt", {Word, Real, Real, Real, Real, Real}];
Close["PuertosATM155ParaGigADSL_LIMA6PE3T0T.txt"];
(*La primera columna es la de los departamentos de LIMA, y las restantes,
el número de puertos desde 2005 a 2009*) PuertosLima = Drop[PuertosLima, 1];
Print["PuertosLima"];
Print[TableForm[PuertosLima]];

Print["NumLIMA"];
NumLIMA = Length[PuertosLima];
Print[NumLIMA];

Print["CabsLIMA"];
CabsLIMA = Table[Plus @@ PuertosLima[[j, k]], {j, NumLIMA}, {k, 2, 6}];
SumCabsLIMA = Plus @@ CabsLIMA;
Print[TableForm[CabsLIMA]];

Print["SumCabsLIMA"];
SumCabsLIMA = Plus @@ CabsLIMA;
Print[SumCabsLIMA];

(*Calcula la demanda GigADSL mayorista
atendida por los PoPs para servicios mayoristas en Lima*)
Print["demADSL"];
demADSL = Table[0, {NumLIMA}];
Do[If[PuertosLima[[j, 1]] == demandasADSL1[[i, 3]], demADSL[[j]] += sumademandasADSL[[i]]];,
  {i, Localidades}, {j, 1, NumLIMA}];
Print[demADSL];

Print["ComprobSumDemADSL"];
ComprobSumDemADSL = Plus @@ demADSL;
Print[ComprobSumDemADSL];

OpenRead["LineasGigADSL.txt"];
LineasGigADSL = ReadList["LineasGigADSL.txt", {Word, Real}];
Close["LineasGigADSL.txt"];
Print["LineasGigADSL"]; Print[TableForm[LineasGigADSL]];

Print[Style["CARGA DE LOS FICHEROS DE ENTRADA",
  Bold, 16, Underlined, White, Background -> Blue]];

(* Antes de Seguir, se Trabajaará con el promedio del promedio móvil de los 5 años*)
(*Print["PromLineasGigADSL"];
PrevLineasGigADSL=Table[(LineasGigADSL[[i-1,2]]+LineasGigADSL[[i,2]])/2,{i,2,5}];
PromLineasGigADSL=Plus@@PrevLineasGigADSL/4;
Print[PromLineasGigADSL];*)
Print["PorcentajeLinGigADSL"];
PorcentajeLinGigADSL =  $\frac{\text{LineasGigADSL}[[1, 2]]}{\text{SumDemADSL}}$ ;
Print[PorcentajeLinGigADSL];
(* Porcentaje de Líneas GigAdsl,
es la cantidad de Líneas entre la demanda ADSL, a nivel anual *)

(*Calculo del numero de puertos ATM donde se consideran PoPs por departamento*)
OpenRead["PuertosATM155ParaGigADSL_Departamentos6PE3T0T.txt"];
PuertosDptos = ReadList[
  "PuertosATM155ParaGigADSL_Departamentos6PE3T0T.txt", {Word, Real, Real, Real, Real, Real}];
Close["PuertosATM155ParaGigADSL_Departamentos6PE3T0T.txt"];
```

```

(*La primera columna es la de los departamentos de Perú, excepto Lima,
y las restantes, el número de puertos desde 2005 a 2009.*)
PuertosDptos = Append[PuertosDptos, Prepend[SumCabsLIMA, "LIMA"]];
Print["PuertosDptos"]; Print[TableForm[PuertosDptos]];

Print["NumDPTOS"];
NumDPTOS = Length[PuertosDptos];
Print[NumDPTOS];

Print[Style["Acumula las Demandas a nivel de Departamento y por año",
  Bold, 16, Underlined, White, Background -> Blue]];

(*Se modifico la expresion para que se acumularan las demandas atendidas por PoPs
a nivel nacional y no limitarlo a la demanda por departamento debido a que los
PoPs ubicados en Arequipa, La Libertad y Lima atienden al resto de departamentos*)

(*Do[Do[If[PuertosDptos[[j,1]]==demandasADSL[[a,i,1]],
  demADSLDep[[a,j]]+=sumademandasADSL[[a,i]]];],
  {i,Localidades},{j,1,NumDPTOS}},{a,Años}]*
Print["demADSLDep"];
demADSLDep = Table[0, {NumDPTOS - 1}];
Do[If[PuertosDptos[[j, 1]] == demandasADSL1[[i, 3]], demADSLDep[[j]] += sumademandasADSL[[i]]];],
  {i, Localidades}, {j, 1, NumDPTOS}]
Print[demADSLDep];
(*Se añadio la siguiente instruccion de suma de comprobacion
de la demanda de usuarios GigADSL mayoristas a nivel nacional*)

Print["demADSLDep"];
demADSLDep = Append[demADSLDep, ComprobSumDemADSL];
Print[demADSLDep];

Print["demADSLDep"];
demADSLDep = Append[demADSLDep, ComprobSumDemADSL];
Print[demADSLDep];

Print["ComprobSumdemADSLDep"];
ComprobSumdemADSLDep = Plus @@ demADSLDep;
Print[ComprobSumdemADSLDep];

Print["CabsDPTOS"];
CabsDPTOS = Table[Plus @@ PuertosDptos[[j, k]], {j, NumDPTOS}, {k, 2, 6}];
Print[TableForm[CabsDPTOS]];

(*Se añadio la siguiente instruccion de suma de comprobacion
del numero de puertos ATM donde se consideran PoPs por departamento*)
Print["ComprobSumCabsDPTOS"];
ComprobSumCabsDPTOS = Table[Plus @@ CabsDPTOS];
Print[ComprobSumCabsDPTOS];

(*Calcula la demanda GigADSL mayorista donde existen puertos ATM a nivel nacional*)
Print["LinGigADSLDep"];
LinGigADSLDep = Table[0, {NumDPTOS}];
Do[Do[If[CabsDPTOS[[j, 1]] ≠ 0, LinGigADSLDep[[j]] += demADSLDep[[j]]], {j, 1, NumDPTOS}]]
Print[LinGigADSLDep];

(* Sumatoria de las líneas ADSL *)
Print["SumLinGigADSLDep"];
SumLinGigADSLDep = Plus @@ LinGigADSLDep;
Print[SumLinGigADSLDep];

(* Se calcula el porcentaje GigADSL por departamento *)
Print["PorcentajesNulos"];
PorcentajesNulos = 0;
Print[PorcentajesNulos];

Print["PorcentGigADSLDep"];
PorcentGigADSLDep =
  If[SumLinGigADSLDep == 0, PorcentajesNulos,  $\frac{\text{SumDemADSL} * \text{PorcentajeLinGigADSL}}{\text{SumLinGigADSLDep}}$ ];
Print[PorcentGigADSLDep];

```

```

(*Calcula la demanda GigADSL mayorista donde existen puertos ATM en el departamento de Lima*)
Print["LineasLimaGigADSL"];
LineasLimaGigADSL = Table[0, {NumLIMA}];
Do[If[CabsLIMA[[j, 1]] ≠ 0, LineasLimaGigADSL[[j]] += demADSL[[j]], {j, 1, NumLIMA}]
Print[LineasLimaGigADSL];

Print["SumLinLimaGigADSL"];
SumLinLimaGigADSL = Plus@@ LineasLimaGigADSL;
Print[SumLinLimaGigADSL];
(*Se modifiko la expresion debido a que en el
porcentaje de lineas GigADSL si bien se toma como dato de entrada,
el valor para cada uno de los años del archivo LineasGigADSL.txt,
este valor es a nivel nacional y esta incluida en la demanda anual pero no especifica que
cantidad realmente corresponde a Lima y que porcentaje para el resto de departamentos
por lo que haciendo los reemplazos corrientes y mas aun con la modificacion del
paso anterior se obtiene el nuevo porcentaje de lineas mayoristas en Lima*)
(*PorcentLimaGigADSL=Table[If[SumLinLimaGigADSL[[a]]==0, PorcentajesNulos[[a]],
SumDemADSL[[a]]*PorcentGigADSLDep[[a]]/SumLinLimaGigADSL[[a]], {a, Años}];*)
Print["PorcentLimaGigADSL"];

PorcentLimaGigADSL = If[SumLinLimaGigADSL == 0, PorcentajesNulos,  $\frac{\text{LineasGigADSL}[[1, 2]]}{\text{SumDemADSL}}$ ];
Print[PorcentLimaGigADSL];

(*Calcula el porcentaje de servicios GigADSL
mayoristas donde existen puertos ATM a nivel nacional*)
Print["PorcentajealneasGigADSLDep"];
PorcentajealneasGigADSLDep =
Table[If[CabsDPTOS[[j, 1]] ≠ 0, PorcentGigADSLDep, 0], {j, 1, NumDPTOS}];
Print[TableForm[PorcentajealneasGigADSLDep]];

(*Calcula el porcentaje de servicios GigADSL mayoristas
donde existen puertos ATM en el departamento de Lima*)
Print["PorcentajealneasGigADSLLima"]; PorcentajealneasGigADSLLima =
Table[If[CabsLIMA[[j, 1]] ≠ 0, PorcentLimaGigADSL, 0], {j, 1, NumLIMA}];
Print[TableForm[PorcentajealneasGigADSLLima]];

Print[Style["CÁLCULO DE LOS COSTES DE NODOS ATM",
Bold, 16, Underlined, White, Background → Blue]];

(*Calcula el numero de usuarios por site para la interfase STM-1*)
Print["UsersSTM1"];
UsersSTM1 =
Table[Min[(maxUsrCluster * InterfazSTM1HaciaPOP[[i]]), sumademandasADSL[[i]]], {i, Localidades}];
Print[TableForm[UsersSTM1]];

Print["SumaUsersSTM1"];
SumaUsersSTM1 = Table[Plus @@ UsersSTM1];
Print[SumaUsersSTM1];

Print["xCx"];
xCx = Table[demandasADSL1[[i, 3]], {i, Localidades}];
Print[TableForm[xCx]];
xCx1 = Length[xCx];
Print[xCx1];

Print["UsersSTM1"];
UsersSTM1 = Table[Prepend[{UsersSTM1[[i]]}, xCx[[i]]], {i, Localidades}];
Print[TableForm[UsersSTM1]];

(*Calcula el numero de usuarios por site para la interfase E3*)
Print["UserE3"];
UserE3 = Table[Min[(maxUsrCluster * InterfazE3HaciaPOP[[i]]),
(sumademandasADSL[[i]] - UsersSTM1[[i, 2]])], {i, Localidades}];
Print[TableForm[UserE3]];

Print["SumUserE3"];
SumUserE3 = Table[Plus @@ UserE3];

```



```

Print[SumUsersE3];

Print["UsersE3"];
UsersE3 = Table[Prepend[{UsersE3[[i]]}, xCx[[i]]], {i, Localidades}];
Print[TableForm[UsersE3]];

(*Calcula el numero de usuarios por site para la interfase E1*)
Print["UsersE1"];
UsersE1 = Table[sumademandasADSL[[i]] - UsersSTM1[[i, 2]] - UsersE3[[i, 2]], {i, Localidades}];
Print[TableForm[UsersE1]];

Print["SumUsersE1"];
SumUsersE1 = Table[Plus@@ UsersE1];
Print[SumUsersE1];

Print["UsersE1"];
UsersE1 = Table[Prepend[{UsersE1[[i]]}, xCx[[i]]], {i, Localidades}];
Print[TableForm[UsersE1]];

(*Calcula el numero de usuarios hacia los Puntos de Presencia (PoPs) a nivel nacional*)
Print["DemPorPOP"];
DemPorPOP = Table[0, {Localidades}];
Do[If[UsersSTM1[[i, 1]] != UsersSTM1[[i - 1, 1]],
  If[UsersSTM1[[j, 1]] == UsersSTM1[[i, 1]], DemPorPOP[[i]] += sumademandasADSL[[j]], 0],
  {i, 2, Localidades}, {j, Localidades}]
Print[DemPorPOP];

Print["DemPorPOP1"];
DemPorPOP1 = Rest[DemPorPOP];
Print[DemPorPOP1];

Print["DemPorPOP2"];
DemPorPOP2 = 0;
Do[If[UsersSTM1[[j, 1]] == UsersSTM1[[1, 1]], DemPorPOP2 += sumademandasADSL[[j]], 0], {j, Localidades}]
Print[DemPorPOP2];

Print["DemPorPOP3"];
DemPorPOP3 = Prepend[DemPorPOP1, DemPorPOP2];
Print[DemPorPOP3];

Print["InterfazSTM1"];
InterfazSTM1 = Table[0, {Localidades}];
Do[If[UsersSTM1[[i, 1]] != UsersSTM1[[i - 1, 1]],
  If[UsersSTM1[[j, 1]] == UsersSTM1[[i, 1]], InterfazSTM1[[i]] += InterfazSTM1HaciaPOP[[j]], 0],
  {i, 2, Localidades}, {j, Localidades}]
Print[InterfazSTM1];

Print["InterfazSTM11"];
InterfazSTM11 = Rest[InterfazSTM1];
Print[InterfazSTM11];

Print["InterfazSTM12"];
InterfazSTM12 = 0;
Do[If[UsersSTM1[[j, 1]] == UsersSTM1[[1, 1]],
  InterfazSTM12 += InterfazSTM1HaciaPOP[[j]], 0], {j, Localidades}]
Print[InterfazSTM12];

(*Calcula el numero de interfaces STM-
1 por cabecera hacia los Puntos de Presencia (PoPs) a nivel nacional*)
Print["InterfazSTM13"];
InterfazSTM13 = Prepend[InterfazSTM11, InterfazSTM12];
Print[InterfazSTM13];

Print["SumInterfazSTM13"];
SumInterfazSTM13 = Plus@@ InterfazSTM13;
Print[SumInterfazSTM13];

Print["InterfazE3"];
InterfazE3 = Table[0, {Localidades}];

```



```

Do[If[UserE3[[i, 1]] ≠ UserE3[[i - 1, 1]],
  If[UserE3[[j, 1]] == UserE3[[i, 1]], InterfazE3[[i]] += InterfazE3HaciaPOP[[j]], 0],
  {i, 2, Localidades}], {j, Localidades}]
Print[InterfazE3];

Print["InterfazE31"];
InterfazE31 = Rest[InterfazE3];
Print[InterfazE31];

Print["InterfazE32"];
InterfazE32 = 0;
Do[
  If[UserE3[[j, 1]] == UserE3[[1, 1]], InterfazE32 += InterfazE3HaciaPOP[[j]], 0], {j, Localidades}];
Print[InterfazE32];

(*Calcula el numero de interfaces E3 por cabecera
 hacia los Puntos de Presencia (PoPs) a nivel nacional*)
Print["InterfazE33"];
InterfazE33 = Prepend[InterfazE31, InterfazE32];
Print[InterfazE33];

Print["SumInterfazE33"];
SumInterfazE33 = Plus @@ InterfazE33;
Print[SumInterfazE33];

Print["InterfazE1"];
InterfazE1 = Table[0, {Localidades}];
Do[If[(UserE1[[i, 1]] ≠ UserE1[[i - 1, 1]]), If[UserE1[[j, 1]] == UserE1[[i, 1]],
  InterfazE1[[i]] += InterfazE1HaciaPOP[[j]], 0], {i, 2, Localidades}], {j, Localidades}];
Print[InterfazE1];

Print["Intfaz1E1"];
Intfaz1E1 = Rest[InterfazE1];
Print[Intfaz1E1];

Print["Intfaz2E1"];
Intfaz2E1 = 0;
Do[If[UserE1[[j, 1]] == UserE1[[1, 1]], Intfaz2E1 += InterfazE1HaciaPOP[[j]], 0], {j, Localidades}]
Print[Intfaz2E1];

(*Calcula el numero de interfaces E1 por cabecera
 hacia los Puntos de Presencia (PoPs) a nivel nacional*)
Print["Intfaz3E1"];
Intfaz3E1 = Prepend[Intfaz1E1, Intfaz2E1];
Print[Intfaz3E1];

Print["SumIntfaz3E1"];
SumIntfaz3E1 = Plus @@ Intfaz3E1;
Print[SumIntfaz3E1];

(*Parámetros de Máximo número de puertos para ADSL: Cluster, Shelf y Tarj*)
OpenRead["Parametros_ATM.txt"];
parametrosATM = ReadList["Parametros_ATM.txt", {Word, Real}];
Print["maxUsrATMCluster"];
maxUsrATMCluster = parametrosATM[[1]][[2]];
Print[maxUsrATMCluster];

Print["maxUsrATMShelf"];
maxUsrATMShelf = parametrosATM[[2]][[2]];
Print[maxUsrATMShelf];

Print["maxUsrATMTarj"];
maxUsrATMTarj = parametrosATM[[3]][[2]];
Print[maxUsrATMTarj];
Close["Parametros_ATM.txt"];
(*Calcula el numero de Clusters en los concentradores ATM por cabecera a nivel nacional*)
Print["ClusterPreconc"];

```

```

ClusterPreconc = Table[Floor[ $\frac{\text{Intfaz3E1}[[i]]}{\text{maxUsrATMCluster}}$ ], {i, Localidades}];
Print[ClusterPreconc];
(*Se ha añadido una instruccion de comprobacion que suma el numero
de Clusters en los concentradores ATM por cabecera a nivel nacional*)

Print["SumClusterPreconc"];
SumClusterPreconc = Plus@@ClusterPreconc;
Print[SumClusterPreconc];
(*Calcula el numero de Shelves en los concentradores ATM por cabecera a nivel nacional*)

Print["ShelfPreconc"];
ShelfPreconc =
  Table[Floor[ $\frac{\text{Intfaz3E1}[[i]] - \text{ClusterPreconc}[[i]] * \text{maxUsrATMCluster}}{\text{maxUsrATMShef}}$ ], {i, Localidades}];
Print[ShelfPreconc];
(*Se ha añadido una instruccion de comprobacion que suma el numero
de Shelves en los concentradores ATM por cabecera a nivel nacional*)

Print["SumShelfPreconc"];
SumShelfPreconc = Plus@@ShelfPreconc;
Print[SumShelfPreconc];
(*Calcula el numero de Tarjetas de Terminacion de
Red en los concentradores ATM por cabecera a nivel nacional*)

Print["TarjPreconc"];
TarjPreconc = Table[
  Ceiling[ $\frac{\text{Intfaz3E1}[[i]] - \text{ClusterPreconc}[[i]] * \text{maxUsrATMCluster} - \text{ShelfPreconc}[[i]] * \text{maxUsrATMShef}}{\text{maxUsrATMTarj}}$ ],
  {i, Localidades}];
Print[TarjPreconc];

(*Se ha añadido una instruccion de comprobacion que suma el numero de Tarjetas
de Terminacion de Red en los concentradores ATM por cabecera a nivel nacional*)

Print["SumTarjPreconc"];
SumTarjPreconc = Plus@@TarjPreconc;
Print[SumTarjPreconc];
(*Calcula el numero de Armarios en los concentradores ATM por cabecera a nivel nacional*)

Print["ArmariosPreconc"];
ArmariosPreconc =
  Table[Ceiling[ $\frac{\text{Intfaz3E1}[[i]] - \text{ClusterPreconc}[[i]] * \text{maxUsrATMCluster}}{2 \text{ maxUsrATMShef}}$ ], {i, Localidades}];
Print[ArmariosPreconc];
(*Se ha añadido una instruccion de comprobacion que suma el numero
de Armarios en los concentradores ATM por cabecera a nivel nacional*)

Print["SumArmariosPreconc"];
SumArmariosPreconc = Plus@@ArmariosPreconc;
Print[SumArmariosPreconc];
(*Calcula el costo del concentrador ATM sin Tarjeta de Terminacion de Red por cabecera*)

Print["CostePreconcSinTarj"];
CostePreconcSinTarj = Table[ClusterPreconc[[i]] * precioPreconcentrClusterFull +
  ShelfPreconc[[i]] * precioPreconcentrShelfFull +
  TarjPreconc[[i]] * precioTarjLT4E1 + ArmariosPreconc[[i]] * precioArmario +
  (If[ShelfPreconc[[i]] > 1, If[TarjPreconc[[i]] > 0, ShelfPreconc[[i]], (ShelfPreconc[[i]] - 1)], 0]) *
  precioShelfExtension + (If[TarjPreconc[[i]] > 0, precioShelf, 0]), {i, Localidades}];
Print[CostePreconcSinTarj];
(*Se ha añadido una instruccion de comprobacion que suma los costos de
los concentradores ATM sin Tarjeta de Terminacion de Red por Cabecera*)

Print["SumCostePreconcSinTarj"];
SumCostePreconcSinTarj = Plus@@CostePreconcSinTarj;
Print[SumCostePreconcSinTarj];
(*Determina el tipo de tarjeta para el Cluster
totalmente ocupado en el concentrador ATM por cabecera*)

```

```

Print["TarjNTPreconcClusterFull"];
TarjNTPreconcClusterFull = Table[If[ClusterPreconc[[i]] > 0, 155, 0], {i, Localidades}];
Print[TarjNTPreconcClusterFull];
(*Se ha a adido una instruccion de comprobacion que suma los tipos de tarjeta para
los Clusters completamente ocupados en los concentradores ATM por cabecera*)
Print["SumTarjNTPreconcClusterFull"];
SumTarjNTPreconcClusterFull = Plus @@ TarjNTPreconcClusterFull;
Print[SumTarjNTPreconcClusterFull];

(*Determina el tipo de tarjeta para el Cluster
parcialmente ocupado en el concentrador ATM por cabecera*)
Print["TarjNTPreconcClusterParcial"];
TarjNTPreconcClusterParcial = Table[If[ShelfPreconc[[i]] > 0, 155,
If[TarjPreconc[[i]] > 0, If[TarjPreconc[[i]] ≤ 4, 34, 155], 0]], {i, Localidades}];
Print[TarjNTPreconcClusterParcial];

(*Se ha a adido una instruccion de comprobacion que suma los tipos de tarjeta
para los Clusters parcialmente ocupados en los concentradores ATM por cabecera*)
Print["SumTarjNTPreconcClusterParcial"];
SumTarjNTPreconcClusterParcial = Plus @@ TarjNTPreconcClusterParcial;
Print[SumTarjNTPreconcClusterParcial];

(*Calcula el costo del concentrador ATM con Tarjeta de Terminacion de Red por cabecera*)
Print["CostePreconcConTarj"];
CostePreconcConTarj =
Table[CostePreconcSinTarj[[i]] + If[TarjNTPreconcClusterFull[[i]] == 34, precioTarjNT34,
If[TarjNTPreconcClusterFull[[i]] == 155, precioTarjNT155, 0]] * ClusterPreconc[[i]] +
If[TarjNTPreconcClusterParcial[[i]] == 34, precioTarjNT34,
If[TarjNTPreconcClusterParcial[[i]] == 155, precioTarjNT155, 0]], {i, Localidades}];
Print[CostePreconcConTarj];

(*Se ha a adido una instruccion de comprobacion que suma los costos de
los concentradores ATM con Tarjeta de Terminacion de Red por cabecera*)
Print["SumCostePreconcConTarj"];
SumCostePreconcConTarj = Plus @@ CostePreconcConTarj;
Print[SumCostePreconcConTarj];

(*Calcula el numero de interfases STM-1 de 155Mbps en el concentrador ATM por cabecera*)
Print["TotIntfzPreconcSTM1"];
TotIntfzPreconcSTM1 = Table[If[TarjNTPreconcClusterFull[[i]] == 155, ClusterPreconc[[i]], 0] +
If[TarjNTPreconcClusterParcial[[i]] == 155, ArmariosPreconc[[i]], 0], {i, Localidades}];
Print[TotIntfzPreconcSTM1];

(*Se ha a adido una instruccion de comprobacion que suma el numero de interfases STM-
1 de 155Mbps en los concentradores ATM por cabecera*)
Print["SumTotIntfzPreconcSTM1"];
SumTotIntfzPreconcSTM1 = Plus @@ TotIntfzPreconcSTM1;
Print[SumTotIntfzPreconcSTM1];

(*Calcula el numero de interfases E3 de 34Mbps en el concentrador ATM por cabecera*)
Print["TotIntfzPreconcE3"];
TotIntfzPreconcE3 = Table[If[TarjNTPreconcClusterFull[[i]] == 34, ClusterPreconc[[i]], 0] +
If[TarjNTPreconcClusterParcial[[i]] == 34, ArmariosPreconc[[i]], 0], {i, Localidades}];
InterfazSTM1HaciaATM = Table[TotIntfzPreconcSTM1[[i]] + InterfazSTM13[[i]], {i, Localidades}];
Print[TotIntfzPreconcE3];

(*Se ha a adido una instruccion de comprobacion que suma el numero
de interfases E3 de 34Mbps en los concentradores ATM por cabecera*)
Print["SumTotIntfzPreconcE3"];
SumTotIntfzPreconcE3 = Plus @@ TotIntfzPreconcE3;
Print[SumTotIntfzPreconcE3];

(*Calcula el numero de interfases STM-1 de 155Mbps en el Nodo ATM por cabecera*)
Print["InterfazSTM1HaciaATM"];
InterfazSTM1HaciaATM = Table[TotIntfzPreconcSTM1[[i]] + InterfazSTM13[[i]], {i, Localidades}];
Print[InterfazSTM1HaciaATM];

(*Se ha a adido una instruccion de comprobacion que suma el numero de interfases STM-

```

```

1 de 155Mbps en los Nodos ATM por cabecera*)
Print["SumInterfazSTM1HaciaATM"];
SumInterfazSTM1HaciaATM = Plus @@ InterfazSTM1HaciaATM;
Print[SumInterfazSTM1HaciaATM];

(*Calcula el numero de interfaces E3 de 34Mbps en el Nodo ATM por cabecera*)
Print["InterfazE3HaciaATM"];
InterfazE3HaciaATM = Table[TotIntfzPreconceE3[[i]] + InterfazE33[[i]], {i, Localidades}];
Print[InterfazE3HaciaATM];

(*Se ha a adido una instruccion de comprobacion que suma el
numero de interfaces E3 de 34Mbps en los Nodos ATM por cabecera*)
Print["SumInterfazE3HaciaATM"];
SumInterfazE3HaciaATM = Plus @@ InterfazE3HaciaATM;
Print[SumInterfazE3HaciaATM];

(*Precios para Nodos ATM en la etapa de concentraci n*)
OpenRead["Precios_ATM.txt"];
preciosATM = ReadList["Precios_ATM.txt", {Word, Real}];
Print["precioChasis"];
precioChasis = preciosATM[[1]][[2]];
Print[precioChasis];
Print["precioTarjSTM1"];
precioTarjSTM1 = preciosATM[[2]][[2]];
Print[precioTarjSTM1];
Print["precioTarjE3"];
precioTarjE3 = preciosATM[[3]][[2]];
Print[precioTarjE3];
Close["Precios_ATM.txt"];

(*Precios para Nodos ATM en la etapa de concentraci n*)
OpenRead["PuertosTarj_ATM.txt"];
puertosTarjATM = ReadList["PuertosTarj_ATM.txt", {Word, Real}];
Print["puertosTarjSTM1"];
puertosTarjSTM1 = puertosTarjATM[[1]][[2]];
Print[puertosTarjSTM1];
Print["puertosTarjE3"];
puertosTarjE3 = puertosTarjATM[[2]][[2]];
Print[puertosTarjE3];
Close["PuertosTarj_ATM.txt"];

(*Calcula el numero de Nodos ATM*)
Print["NumNodosATM"];
NumNodosATM = Table[Ceiling[ $\frac{\text{DemPorPOP3}[[i]]}{16\,000}$ ], {i, Localidades}];
Print[NumNodosATM];

(*Se ha a adido una instruccion de comprobacion
que suma el numero de Nodos ATM por cabecera*)
Print["SumNumNodosATM"];
SumNumNodosATM = Plus @@ NumNodosATM;
Print[SumNumNodosATM];

(*Calcula el numero de tarjetas STM-1 en el Nodo ATM por cabecera*)
Print["NumTarjSTM1Conc"];
NumTarjSTM1Conc = Table[Ceiling[ $\frac{\text{InterfazSTM1HaciaATM}[[i]]}{\text{puertosTarjSTM1}}$ ], {i, Localidades}];
Print[NumTarjSTM1Conc];

(*Se ha a adido una instruccion de comprobacion que suma el numero de Tarjetas STM-
1 en los Nodos ATM por cabecera*)
Print["SumNumTarjSTM1Conc"];
SumNumTarjSTM1Conc = Plus @@ NumTarjSTM1Conc;
Print[SumNumTarjSTM1Conc];

(*Calcula el numero de Tarjetas E3 en el Nodo ATM por cabecera*)
Print["NumTarjE3Conc"];
NumTarjE3Conc = Table[Ceiling[ $\frac{\text{InterfazE3HaciaATM}[[i]]}{\text{puertosTarjE3}}$ ], {i, Localidades}];

```

```

Print[NumTarjE3Conc];

(*Se ha a adido una instruccion de comprobacion que
suma el numero de Tarjetas E3 en los Nodos ATM por cabecera*)
Print["SumNumTarjE3Conc"];
SumNumTarjE3Conc = Plus@@ NumTarjE3Conc;
Print[SumNumTarjE3Conc];

(*Calcula el costo del Nodo ATM por cabecera*)
Print["CostoATMConc"];
CostoATMConc = Table[NumNodosATM[[i]] * precioChasis +
    NumTarjSTM1Conc[[i]] * precioTarjSTM1 + NumTarjE3Conc[[i]] * precioTarjE3, {i, Localidades}];
Print[CostoATMConc];

(*Se ha a adido una instruccion de comprobacion
que suma los costos de los Nodos ATM por cabecera*)
Print["SumCostoATMConc"];
SumCostoATMConc = Plus@@ CostoATMConc;
Print[SumCostoATMConc];

(*Carga los datos de n mero de usuarios*)
OpenRead["NumUsuarios_Agreg.txt"];
NumUsersAgreg = ReadList["NumUsuarios_Agreg.txt", {Word, Real}];
Print["NumUsersSTM1Agreg"];
NumUsersSTM1Agreg = NumUsersAgreg[[1]][[2]];
Print[NumUsersSTM1Agreg];

Print["NumUsersAgregAgreg"];
NumUsersAgregAgreg = NumUsersAgreg[[2]][[2]];
Print[NumUsersAgregAgreg];
Close["NumUsuarios_Agreg.txt"];

Print["yCx"];
yCx = Table[demandasADSL1[[i, 1]], {i, Localidades}];
Print[TableForm[yCx]];
yCx1 = Length[yCx];
Print[yCx1];

Print["DemPorPOP4"];
DemPorPOP4 = Prepend[DemPorPOP3, xCx];
Print[DemPorPOP4];

Print["DemPorPOP5"];
DemPorPOP5 = Prepend[DemPorPOP4, yCx];
Print[DemPorPOP5];

Print["z1Cx"];
z1Cx = 0;
Print[z1Cx];
(*En formato normal instrucciones que se han a adido al programa*)
Print["PorcentajealneasGigADSLLima2"];
PorcentajealneasGigADSLLima2 = PorcentajealneasGigADSLLima;
Print[TableForm[PorcentajealneasGigADSLLima2]];

Print["PorclneasGigADLSLima2"];
PorclneasGigADLSLima2 = Length[PorcentajealneasGigADSLLima];
Print[PorclneasGigADLSLima2];

Print["PorcentajealneasGigADSLLima2"];
Do[PorcentajealneasGigADSLLima2 = Table[Append[PorcentajealneasGigADSLLima2, z1Cx]],
    {1, NumDPTOS - NumLIMA}]
Print[TableForm[PorcentajealneasGigADSLLima2]];

Print["PorclneasGigADLSLima22"];
PorclneasGigADLSLima22 = Length[PorcentajealneasGigADSLLima2];
Print[PorclneasGigADLSLima22];
PorcentajealneasGigADSLLima2;

Print["z2Cx"];

```



```

z2Cx = Table[0, {6}];
Print[z2Cx];

Print["PuertosLima2"];
PuertosLima2 = PuertosLima;
Print[TableForm[PuertosLima2]];

Print["PtosLima2"];
PtosLima2 = Length[PuertosLima];
Print[PtosLima2];

Print["PuertosLima2"];
Do[PuertosLima2 = Table[Append[PuertosLima2, z2Cx]], {l, NumDPTOS - NumLIMA}]
Print[TableForm[PuertosLima2]];

Print["PtosLima22"];
PtosLima22 = Length[PuertosLima];
Print[PtosLima22];

Print["PorcentajeGigADSLTotal1"];
PorcentajeGigADSLTotal1 =
  Table[If[DemPorPOP5[[1, i]] ≠ "LIMA", If[DemPorPOP5[[1, i]] == PuertosDptos[[j, 1]],
    PorcentajealneasGigADSLDep[[j]], 0], If[DemPorPOP5[[2, i]] == PuertosLima2[[j, 1]],
    PorcentajealneasGigADSLLima2[[j]], 0]], {i, Localidades}, {j, NumDPTOS}];
Print[PorcentajeGigADSLTotal1];

(*Calcula el porcentaje de lineas mayoristas por site a nivel nacional*)
Print["PorcentajeGigADSLTotal"];
PorcentajeGigADSLTotal = Table[Plus @@ PorcentajeGigADSLTotal1[[i]], {i, Localidades}];
Print[TableForm[PorcentajeGigADSLTotal]];

(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*NumInterSTM1Agreg=Table[
  Ceiling[DemPorPOP3[[i]]*(1-PorcentajeGigADSLTotal[[i]])/NumUsersSTM1Agreg], {i, Localidades}];
Print["NumInterSTM1Agreg"];
Print[NumInterSTM1Agreg];
SumComprobNumInterSTM1Agreg=Plus @@ NumInterSTM1Agreg;
Print[SumComprobNumInterSTM1Agreg];*)

(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*NumInterSTM1Agreg1=Prepend[NumInterSTM1Agreg,xCx];
Print["NumInterSTM1Agreg1"];
Print[NumInterSTM1Agreg1];*)

(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Calcula el numero de interfases STM-1 para los servicios minoristas por cabecera*)
(*NumInterSTM1Agreg2=Prepend[NumInterSTM1Agreg1,yCx];
Print["NumInterSTM1Agreg2"];
Print[NumInterSTM1Agreg2];*)

(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Guarda los resultados obtenidos en NumInterSTM1Agreg2*)
PRUPtasATM = OpenWrite["PRUPtasATM.txt"];
WriteString[PRUPtasATM, "", "\n", "\n", "Dptos",
  " ", AccountingForm[NumInterSTM1Agreg2[[1]], "\n", "Cabs", " ",
  AccountingForm[NumInterSTM1Agreg2[[2]], "\n", "03", " ", AccountingForm[NumInterSTM1Agreg2[[3]],
  "\n", "04", " ", AccountingForm[NumInterSTM1Agreg2[[4]], "\n", "05", " ",
  AccountingForm[NumInterSTM1Agreg2[[5]], "\n", "06", " ", AccountingForm[NumInterSTM1Agreg2[[6]],
  "\n", "07", " ", AccountingForm[NumInterSTM1Agreg2[[7]], "\n"];
Close["PRUPtasATM.txt"];

(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*PuertosATMGigADSL1=Table[If[NumInterSTM1Agreg2[[1,i]]≠"LIMA",
  If[NumInterSTM1Agreg2[[1,i]]==PuertosDptos[[j,1]],PuertosDptos[[j,a+1]],0],

```

```

    If[NumInterSTM1Agreg2[[2,i]]==PuertosLima2[[j,1]],PuertosLima2[[j,a+1],0]],
    {i,Localidades},{a,Años},{j,NumDPTOS}];
Print["PuertosATMGigADSL1"];
Print[PuertosATMGigADSL1];*)
(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Calcula el numero de puertos ATM para los servicios mayoristas por site*)
(*PuertosATMGigADSL=Table[Plus@@PuertosATMGigADSL1[[i,a]],{a,Años},{i,Localidades}];
Print["PuertosATMGigADSL"];
Print[PuertosATMGigADSL];

Print["SumComprobPuertosATMGigADSL"];
SumComprobPuertosATMGigADSL=
    Table[Plus@@PuertosATMGigADSL[[a]],{a,Años}];
Print[SumComprobPuertosATMGigADSL];*)
(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Cálcula la suma del número de interfases y puertos STM-1 hacia los PoPs ATM*)
(*NumPuertasATM=Table[If[NumInterSTM1Agreg[[a,i]]≠0,
    NumInterSTM1Agreg[[a,i]]+PuertosATMGigADSL[[a,i]],0],{a,Años},{i,Localidades}];
Print["NumPuertasATM"];
Print[NumPuertasATM];
Print["ComprobNumPuertasATM"];
ComprobNumPuertasATM=Table[ Plus@@NumPuertasATM[[a]],{a,Años}];
Print[ComprobNumPuertasATM];*)
(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Calcula el costo total del Nodo ATM por cabecera*)
(*CostoNodoATMCnxOp=
    Table[Ceiling[NumPuertasATM[[a,i]]/4]*precioTarjSTM1,{a,Años},{i,Localidades}];
Print["CostoNodoATMCnxOp"];
Print[CostoNodoATMCnxOp];
Print["SumComprobCostoNodoATMCnxOp"];
SumComprobCostoNodoATMCnxOp=
    Table[Plus@@CostoNodoATMCnxOp[[a]],{a,Años}];
Print[SumComprobCostoNodoATMCnxOp];*)
(*No se esta considerando la parte de los Agregadores IP. Se
va a desechar del modelo final por considerarlo que se añadiendo
un costo que no corresponde en el calculo del costo del Nodo ATM*)
(*Calcula el costo total del Punto de Presencia (PoP) ATM por cabecera*)
(*CostoTotalPopATM=Table[CostePreconcConTarj[[a,i]]+
    CostoATMConc[[a,i]]+CostoNodoATMCnxOp[[a,i]],{a,Años},{i,Localidades}];*)
CostoTotalPopATM = Table[CostePreconcConTarj[[i]] + CostoATMConc[[i]], {i, Localidades}];

Print["CostoTotalPopATM"];
Print[CostoTotalPopATM];

Print["SumCostoTotalPopATM"];
SumCostoTotalPopATM = Plus @@ CostoTotalPopATM;
Print[SumCostoTotalPopATM];

(*Carga los datos de potencia y energía*)
OpenRead["Energia_ATM.txt"];
energiaATM = ReadList["Energia_ATM.txt", {Word, Real}];
Close["Energia_ATM.txt"];

Print["CostePorWATM"];
CostePorWATM = energiaATM[[1]][[2]];
Print[CostePorWATM];

Print["WPorEquipoATM"];
WPorEquipoATM = energiaATM[[2]][[2]];
Print[WPorEquipoATM];

(*Calcula el costo de energia del Nodo ATM*)
Print["CostoEnergiaATM"];

```

```
CostoEnergiaATM = Table[CostePorWATM * WPorEquipoATM * NumNodosATM[[i]], {i, Localidades}];
Print[CostoEnergiaATM];
```

```
Print["SumCostoEnergiaATM"];
SumCostoEnergiaATM = Plus @@ CostoEnergiaATM;
Print[SumCostoEnergiaATM];
```

```
(***** Fin de la
parte de Conmutación. *****)
```

```
Print[Style["PARÁMETROS COSTES", Bold, 16, Underlined, White, Background -> Blue],
```

```
(*Carga los parámetros de costes a emplear en la ejecución*)
OpenRead["Parametros_Costes.txt"];
ParamsCostes = ReadList["Parametros_Costes.txt", {Word, Real}];
Wacc = ParamsCostes[[1]][2];
AñosAmortCxDSLAM = ParamsCostes[[2]][2];
AñosAmortCxNodoATM = ParamsCostes[[3]][2];
AñosAmortTxEquipos = ParamsCostes[[4]][2];
AñosAmortTxPtaExt = ParamsCostes[[5]][2];
AñosAmortTxPortador = ParamsCostes[[6]][2];
AñosAmortPtaSecENERGIA = ParamsCostes[[7]][2];
AñosAmortPtaSecINMOB = ParamsCostes[[8]][2];
```

```
Close["Parametros_Costes.txt"];
(*Puesto que el valor del WACC es después de impuestos,
hay que convertirlo a su valor antes de impuestos*)
TasaImpositiva = 0.37;
```

```
CosteCapital =  $\frac{Wacc}{1 - TasaImpositiva}$ ;
```

```
(*Una vez introducidos los datos de entrada,
se calculan los respectivos factores de anualización*)
```

```
FactAnualizCxDSLAM =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortCxDSLAM}}$ ;
```

```
FactAnualizCxNodoATM =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortCxNodoATM}}$ ;
```

```
FactAnualizTxEquipos =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortTxEquipos}}$ ;
```

```
FactAnualizTxPtaExt =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortTxPtaExt}}$ ;
```

```
FactAnualizTxPortador =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortTxPortador}}$ ;
```

```
FactAnualizPtaSecENERGIA =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortPtaSecENERGIA}}$ ;
```

```
FactAnualizPtaSecINMOB =  $\frac{CosteCapital}{1 - (1 + CosteCapital)^{-AñosAmortPtaSecINMOB}}$ ;
```

```
Print[Style["EVOLUCIÓN PRECIOS", Bold, 16, Underlined, White, Background -> Blue],
```

```
(*Carga del fichero que determina la evolución de precios*)
OpenRead["EvolucionPrecios.txt"];
EvPrecios = ReadList["EvolucionPrecios.txt", {Word, Real, Real, Real, Real, Real}];
EvPrecios = Drop[EvPrecios, 1];
Close["EvolucionPrecios.txt"];
```

```
(*Carga del fichero que determina los costes de actividades para cada año*)
OpenRead["CosteActividadesCambios.txt"];
CosteActivs2 = ReadList["CosteActividadesCambios.txt", {Word, Real, Real, Real, Real, Real}];
CosteActivs2 = Drop[CosteActivs2, 1];
Close["CosteActividadesCambios.txt"];
```

```
OpenRead["PorcentCosteActiv.txt"];
CosteActivs = ReadList["PorcentCosteActiv.txt", {Word, Real}];
Close["PorcentCosteActiv.txt"];
```

```
Print["CADSL"];
CADSL = Table[0, {ix, nx}];
```

```

Do[If[EquivADSLFija[[ix]] ≠ 0,
  auxLoc = EquivADSLFija[[ix]];
CADSL[[ix]] = (63 * InterfazSTM1HaciaPOP[[auxLoc]] +
  16 * InterfazE3HaciaPOP[[auxLoc]] + InterfazE1HaciaPOP[[auxLoc]])];, {ix, nx}];
Print[CADSL];
(*Cargas en número de E1s de ADSL por cada central*)

Print["CADSLd"];
CADSLd = Table[0, {id, nd}];
Do[CADSLd[[deptx[[ix]]]] = CADSLd[[deptx[[ix]]]] + CADSL[[ix]], {ix, nx}];
Print[CADSLd];

Print["CADSLNucd"];
CADSLNucd = Table[0, {id, nd}];
Do[CADSLNucd[[deptx[[ix]]]] = CADSLNucd[[deptx[[ix]]]] + NumE1EquivsNuc[[ix]]; , {ix, nx}];
Print[CADSLNucd];
(* Acumulación de las cargas por departamento *)

inputLDadsl = Table[0, {id, nd}, {jd, nd}];
Destinosadsl = Table[0, {id, nd}];

inputLDNucNuc = Table[0, {id, nd}, {jd, nd}];
adslTruji = 0;
adslAreq = 0;
Do[Destinosadsl[[id]] =
Switch[id, AYACUCHO, LIMA,
  HUANCVELICA, LIMA,
  HUANUCO, LIMA,
  ICA, LIMA,
  JUNIN, LIMA,
  LIMA, LIMA,
  LORETO, LIMA,
  PASCO, LIMA,
  SANMARTIN, LALIBERTAD,
  UCAYALI, LIMA,
  AMAZONAS, LALIBERTAD,
  ANCASH, LALIBERTAD,
  CAJAMARCA, LALIBERTAD,
  LALIBERTAD, LALIBERTAD,
  LAMBAYEQUE, LALIBERTAD,
  PIURA, LALIBERTAD,
  TUMBES, LALIBERTAD,
  APURIMAC, AREQUIPA,
  AREQUIPA, AREQUIPA,
  CUSCO, AREQUIPA,
  MADREDEDIOS, LIMA,
  MOQUEGUA, AREQUIPA,
  PUNO, AREQUIPA,
  TACNA, AREQUIPA];
  If[Destinosadsl[[id]] == 14, adslTruji = adslTruji + CADSLNucd[[id]];
  If[Destinosadsl[[id]] == 19, adslAreq = adslAreq + CADSLNucd[[id]];
  inputLDadsl[[id, Destinosadsl[[id]]]] = inputLDadsl[[id, Destinosadsl[[id]]]] + CADSLd[[id]];
, {id, nd}];
inputLDNucNuc[[14, 6]] = Ceiling[Ceiling[adslTruji], 63];
inputLDNucNuc[[19, 6]] = Ceiling[Ceiling[adslAreq], 63];

Print[inputLDadsl];
Print[inputLDNucNuc];
(* Se ingresan los archivos de Cargas *)
Print["input_traficoLDadsl.txt"];
Print["input_traficoAlqLDE1sABC2.txt"];
OutAdslLD = OpenWrite["input_traficoLDadsl.txt"];
OutNucNucLD = OpenWrite["input_traficoAlqLDE1sABC2.txt"];
Do[str1 = "";
  str2 = "";
  Do[If[iy == 1,
    str1 = StringJoin[{str1, ToString[inputLDadsl[[ix, iy]]}]];
    str2 = StringJoin[{str2, ToString[inputLDNucNuc[[ix, iy]]}]];

```

```
str1 = StringJoin[{str1, "\t", ToString[inputLDadsl[[ix, iy]]]};
str2 = StringJoin[{str2, "\t", ToString[inputLDNucNuc[[ix, iy]]]};];];, {iy, 24}];
str1 = StringJoin[{str1, "\n"}];
str2 = StringJoin[{str2, "\n"}];
WriteString[OutAdslLD, str1];
WriteString[OutNucNucLD, str2];, {ix, 24}];
Close[OutAdslLD];
Close[OutNucNucLD];
(* Se acumula aquí los tráficos ADSL por TRAMO *)

Print["nadsl"];
nadsl = Length[CADSL];
Print[nadsl];

Print["TOTALE1ADSL"];
TOTALE1ADSL = 0;
Do[TOTALE1ADSL = TO TALE1ADSL + CADSL[[iadsl]], {iadsl, nadsl}];
Print[TOTALE1ADSL];

e1remotehostadslx = zerox;
e1hosttandemadslx = zerox;

Print["e1remotehostadslx"];
Do[
  If[remote1x[[iadsl]] == 1, e1remotehostadslx[[iadsl]] = e1remotehostadslx[[iadsl]] + CADSL[[iadsl]];];,
  {iadsl, nadsl}];
Print[e1remotehostadslx]

Print["e1hosttandemadslx"];
Do[If[remote1x[[iadsl]] == 1,
  e1hosttandemadslx[[allhostx[[iadsl]]]] = e1hosttandemadslx[[allhostx[[iadsl]]]] + CADSL[[iadsl]];];
  If[remote1x[[iadsl]] == 0 & tandem1x[[iadsl]] == 0,
    e1hosttandemadslx[[iadsl]] = e1hosttandemadslx[[iadsl]] + CADSL[[iadsl]];];, {iadsl, nadsl}];

Do[If[limax[[iadsl]] & esAnillodx[[iadsl]] == 2 & hostx[[iadsl]], If[CxATMLimax[[iadsl]] != 1,
  e1hosttandemadslx[[iadsl]] = Ceiling[e1hosttandemadslx[[iadsl]] * 1.75];,
  e1hosttandemadslx[[iadsl]] = Ceiling[e1hosttandemadslx[[iadsl]] * 0.75]];];, {iadsl, nadsl}];
(*Esta instucción sirve para utilizar el 75% más de Tx para ADSL en anillo de Lima*)
Print[e1hosttandemadslx];
```

```
Print[
  Style["ESTIMACIÓN DE CARGAS EN E1S ASOCIADAS A LOS CIRCUITOS ALQUILADOS  

Y QUE SON IMPUTABLES A LAS REDES INTRADEPARTAMENTALES",
    Bold, 16, Underlined, White, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print[Style["CIRCUITOS RANGO A", Bold, 16, Underlined, White, Background -> Blue]];
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES",
  Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_AlquiladosA1.txt"];
(* Leyendo la información de los circuitos alquilados del rango A a nivel Interdepartamental:
  La data viene así:
  39      1
  377     1
  377     1
  El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál),
  el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado
  con alguna de las centrales del modelo, en el ejemplo, los números 39 y 377 hacen
  referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces,
  en una fila aparecera el origen o destino y en otra el correspondiente destino u origen,
  sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no
  es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

Print["nalqA1"];
CALQUILADOA1 = Traducir[N[ReadList["input_C_AlquiladosA1.txt", Number, RecordLists -> True]]];
nalqA1 = Length[CALQUILADOA1];
Print[nalqA1];

Print["CALQUILADOA1"];
Print[TableForm[Table[{CALQUILADOA1[[ialq, 1]], y[[CALQUILADOA1[[ialq, 1]], 13]],
  deptTXTd[deptx[[CALQUILADOA1[[ialq, 1]]]], CALQUILADOA1[[ialq, 2]]}, {ialq, nalqA1}], TableSpacing -> {0, 1},
  TableHeadings -> {None, {"ix", "Nombre Central", "Departamento", "Nro Circuitos"}}]]];
```



```

(* Presenta la información del archivo de entrada de los circuitos alquilados del rango
A a nivel Interdepartamental como una lista donde cada elemento es a su vez una
lista que presenta los datos de cada fila: [ [origen o destino , número de E1s] ] *)

(* Identifica el número de elementos de la lista CALQUILADOA1 *)

e1remotehostalqxA1 = zerox;
e1hosttandemalqxA1 = zerox;
TOTALE1ALQA1 = 0;
Print["TOTALE1ALQA1"];

Do[
$$\text{TOTALE1ALQA1} = \text{TOTALE1ALQA1} + \frac{\text{CALQUILADOA1}[[\text{ialq}, 2]]}{2};, \{\text{ialq}, \text{nalqA1}\}];$$

Print[TOTALE1ALQA1];
(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo A que son atribuibles como
cargas en el tramo intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista),
se suman todos los E1s identificados (2do elemento de cada sublista). El
resultado se divide entre 2 porque cada circuito aparece dos veces *)

Print["e1remotehostalqxA1"];
Do[If[remote1x[[CALQUILADOA1[[ialq, 1]]] == 1, e1remotehostalqxA1[[CALQUILADOA1[[ialq, 1]]] =
e1remotehostalqxA1[[CALQUILADOA1[[ialq, 1]]] + CALQUILADOA1[[ialq, 2]]];, {ialq, nalqA1}];
Print[e1remotehostalqxA1];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del
tipo A que estan asociados como punto de oriden o destino a cada central remota. Dentro
de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista),
identifica las centrales que son remotas y para cada una de ellas realiza
la sumatoria del total de E1s (2do elemento de cada sublista) *)

Print["e1hosttandemalqxA1"];
Do[If[remote1x[[CALQUILADOA1[[ialq, 1]]] == 1, e1hosttandemalqxA1[[allhostx[[CALQUILADOA1[[ialq, 1]]]]] =
e1hosttandemalqxA1[[allhostx[[CALQUILADOA1[[ialq, 1]]]]] + CALQUILADOA1[[ialq, 2]]];
If[remote1x[[CALQUILADOA1[[ialq, 1]]] == 0 ^ tandem1x[[CALQUILADOA1[[ialq, 1]]] == 0,
e1hosttandemalqxA1[[CALQUILADOA1[[ialq, 1]]] =
e1hosttandemalqxA1[[CALQUILADOA1[[ialq, 1]]] + CALQUILADOA1[[ialq, 2]]];, {ialq, nalqA1}];
Print[e1hosttandemalqxA1];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales
del tipo A que deberán ser añadidos a las cargas a nivel de centrales Host. Se
realiza lo siguiente: En la primera parte del código identifica las centrales
host que son cabeceras de las remotas en las cuales se identificó circuitos asociados,
y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas
(por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista
de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem
y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_Alquilados2A1.txt"];
(* Leyendo la información de los circuitos alquilados del rango A que son intradepartamentales:
La data viene así:
110    120    1
248    261    1
248    347    1
El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de EIs *)

Print["CALQUILADOA12"];
CALQUILADOA12 = Traducir[N[ReadList["input_C_Alquilados2A1.txt", Number, RecordLists -> True]]];
Print[CALQUILADOA12];
(* Presenta la información del archivo de entrada de los circuitos alquilados
del rango A a nivel Intradepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [ [origen, destino, número de E1s] ] *)

Print["nalqA12"];
nalqA12 = Length[CALQUILADOA12];
Print[nalqA12];
(* Identifica el número de elementos de la lista CALQUILADOA12*)

Print["TOTALE1ALQA1"];
Do[TOTALE1ALQA1 = TOTALE1ALQA1 + CALQUILADOA12[[ialq, 3]];, {ialq, nalqA12}];
Print[TOTALE1ALQA1];
(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo A. Se suman todos los E1s
considerados como data (3er elemento de cada sublista). Nótese que dichos resultados se estan acumulando con la
estimación ya realizada anteriormente para la variable TOTALE1ALQA1 que correspondió a los E1s asociados a los
circuitos Interdepartamentales del tipo A que eran atribuibles como cargas en el tramo intradepartamental *)

Print["e1remotehostalqxA1"];
Do[aux1 = CALQUILADOA12[[ialq, 1]];
aux2 = CALQUILADOA12[[ialq, 2]];
If[aux1 != aux2,
If[remote1x[[aux1]] == 1, e1remotehostalqxA1[[aux1]] = e1remotehostalqxA1[[aux1]] + CALQUILADOA12[[ialq, 3]]];

```

```
If[remote1x[[aux2]] == 1, e1remotehostalqxA1[[aux2]] = e1remotehostalqxA1[[aux2]] + CALQUILADOA12[[ialq, 3]]];];
, {ialq, nalqA12}];
Print[e1remotehostalqxA1];
(* Esta identificando los E1s correspondientes a circuitos Intradepartamentales del
tipo A que estan asociados como punto de oriden o destino a cada central remota. Se hace lo
siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que
son remotas y se suma para cada una de ellas el total de E1s (3er elemento de cada sublista). La segunda
parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se
suma para cada una de ellas el total de E1s (3er elemento de cada sublista). Nótese que dichos resultados
se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqxA1 *)

Print["e1hosttandemalqxA1"]
Do[aux1 = CALQUILADOA12[[ialq, 1]];
aux2 = CALQUILADOA12[[ialq, 2]];
If[aux1 != aux2,
(* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]]];
If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]]];
(* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
en caso de no ser remota es claro que el nodo es identificado como Host *)
If[tandem1x[[aux2]] == 0 & aux1 != aux2,
e1hosttandemalqxA1[[aux2]] = e1hosttandemalqxA1[[aux2]] + CALQUILADOA12[[ialq, 3]]];
(* A nivel de las centrales de destino, Si la cabecera del destino no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del origen,
se suma para cada una de las cabeceras del destino el número de E1s de sus remotas identificadas como
destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
If[tandem1x[[aux1]] == 0 & aux1 != aux2,
e1hosttandemalqxA1[[aux1]] = e1hosttandemalqxA1[[aux1]] + CALQUILADOA12[[ialq, 3]]];];
, {ialq, nalqA12}]
(* A nivel de las centrales de origen, Si la cabecera del origen no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del destino,
se suma para cada una de las cabeceras del origen el número de E1s de sus remotas identificadas como
origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
Print[e1hosttandemalqxA1];
(* Este desarrollo esta identificando los E1s correspondientes a circuitos Intradepartamentales del tipo
A que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados se
estan acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxA1 *)

Print[Style["CIRCUITOS RANGO B", Bold, 16, Underlined, White, Background -> Blue]];
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES",
Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_AlquiladosB1.txt"];
(* Leyendo la información de los circuitos alquilados del rango B a nivel Interdepartamental:
La data viene así:
39 1
377 1
377 1
El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál),
el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado
con alguna de las centrales del modelo, en el ejemplo, los números 39 y 377 hacen
referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces,
en una fila aparecera el origen o destino y en otra el correspondiente destino u origen,
sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no
es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

Print["CALQUILADOB1"];
CALQUILADOB1 = Traducir[N[ReadList["input_C_AlquiladosB1.txt", Number, RecordLists -> True]]];
Print[CALQUILADOB1];
(* Presenta la información del archivo de entrada de los circuitos alquilados
del rango B a nivel Interdepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [[origen o destino,número de E1s]] *)

Print["nalqB1"];
nalqB1 = Length[CALQUILADOB1];
Print[nalqB1];
(* Identifica el número de elementos de la lista CALQUILADOB1 *)

e1remotehostalqxB1 = zerox;
e1hosttandemalqxB1 = zerox;
TOTALE1ALQB1 = 0;
Print["TOTALE1ALQB1"];
Do[TOTALE1ALQB1 = TOTALE1ALQB1 +  $\frac{CALQUILADOB1[[ialq, 2]]}{2}$ ; , {ialq, nalqB1}];
Print[TOTALE1ALQB1];
(* Identifica el total de E1s asociados a los circuitos Interdepartamentales
del tipo B que son atribuibles como cargas en el tramo intradepartamental. Se realiza
```

lo siguiente: independientemente del nodo de referencia (1er dato de cada sublista), se suman todos los E1s identificados (2do elemento de cada sublista). El resultado se divide entre 2 porque cada circuito aparece dos veces *)

```
Print["e1remotehostalqxB1"];
```

```
Do[If[remote1x[[CALQUILADOB1[[ialq, 1]]] == 1, e1remotehostalqxB1[[CALQUILADOB1[[ialq, 1]]] =  
    e1remotehostalqxB1[[CALQUILADOB1[[ialq, 1]]] + CALQUILADOB1[[ialq, 2]]];], {ialq, nalqB1}];  
Print[e1remotehostalqxB1];
```

(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo B que estan asociados como punto de oriden o destino a cada central remota. Dentro de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista), identifica las centrales que son remotas y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print["e1hosttandemalqxB1"];
```

```
Do[If[remote1x[[CALQUILADOB1[[ialq, 1]]] == 1, e1hosttandemalqxB1[[allhostx[[CALQUILADOB1[[ialq, 1]]]]] =  
    e1hosttandemalqxB1[[allhostx[[CALQUILADOB1[[ialq, 1]]]]] + CALQUILADOB1[[ialq, 2]]];]  
If[remote1x[[CALQUILADOB1[[ialq, 1]]] == 0 & tandem1x[[CALQUILADOB1[[ialq, 1]]] == 0,  
    e1hosttandemalqxB1[[CALQUILADOB1[[ialq, 1]]] =  
    e1hosttandemalqxB1[[CALQUILADOB1[[ialq, 1]]] + CALQUILADOB1[[ialq, 2]]];], {ialq, nalqB1}];  
Print[e1hosttandemalqxB1];
```

(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo B que deberán ser añadidos a las cargas a nivel de centrales Host. Si se optó por la Opción 3 o 4 de estimar los costos de los circuitos del tipo A o B se realiza lo siguiente: En la primera parte del código identifica las centrales host que son cabeceras de las remotas en las cuales se identificó circuitos asociados, y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas (por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background → Yellow]];
```

```
OpenRead["input_C_Alquilados2B1.txt"];
```

(* Leyendo la información de los circuitos alquilados del rango B que son intradepartamentales:

La data viene así:

```
377    259    1
```

```
39     59     1
```

```
248    252    1
```

El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de E1s *)

```
Print["CALQUILADOB12"];
```

```
CALQUILADOB12 = Traducir[N[ReadList["input_C_Alquilados2B1.txt", Number, RecordLists → True]]];
```

```
Print[CALQUILADOB12];
```

(* Presenta la información del archivo de entrada de los circuitos alquilados del rango B a nivel Intradepartamental como una lista donde cada elemento es a su vez una lista que presenta los datos de cada fila: [[origen,destino,número de E1s]] *)

```
Print["nalqB12"];
```

```
nalqB12 = Length[CALQUILADOB12];
```

```
Print[nalqB12];
```

(* Identifica el número de elementos de la lista CALQUILADOB12 *)

```
Print["TOTALE1ALQB1"];
```

```
Do[TOTALE1ALQB1 = TOTALE1ALQB1 + CALQUILADOB12[[ialq, 3]];, {ialq, nalqB12}];
```

```
Print[TOTALE1ALQB1];
```

(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo B. Se suman todos los E1s considerados como data (3er elemento de cada sublista). Nótese que dichos resultados se estan acumulando con la estimación ya realizada anteriormente para la variable TOTALE1ALQB1 que correspondió a los E1s asociados a los circuitos Interdepartamentales del tipo A que eran atribuibles como cargas en el tramo intradepartamental *)

```
Print["e1remotehostalqxB1"];
```

```
Do[aux1 = CALQUILADOB12[[ialq, 1]];
```

```
    aux2 = CALQUILADOB12[[ialq, 2]];
```

```
    If[aux1 ≠ aux2,
```

```
        If[remote1x[[aux1]] == 1, e1remotehostalqxB1[[aux1]] = e1remotehostalqxB1[[aux1]] + CALQUILADOB12[[ialq, 3]]];
```

```
        If[remote1x[[aux2]] == 1, e1remotehostalqxB1[[aux2]] = e1remotehostalqxB1[[aux2]] + CALQUILADOB12[[ialq, 3]]];];
```

```
    , {ialq, nalqB12}];
```

```
Print[e1remotehostalqxB1];
```

(* Esta identificando los E1s correspondientes a circuitos Intradepartamentales del tipo B que estan asociados como punto de oriden o destino a cada central remota. Se hace lo siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que son remotas y se suma para cada una de ellas el total de E1s (3er elemento de cada sublista). La segunda parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se suma para cada una de ellas el total de E1s (3er elemento de cada sublista). Nótese que dichos resultados se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqxB1 *)

```
Print["e1hosttandemalqxB1"];
```

```
Do[aux1 = CALQUILADOB12[[ialq, 1]];
```

```

aux2 = CALQUILADOB12[[ialq, 2]];
If[aux1 ≠ aux2,
  (* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
  If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]];];
  If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]];];
  (* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
  en caso de no ser remota es claro que el nodo es identificado como Host *)
  If[tandem1x[[aux2]] == 0 ∧ aux1 ≠ aux2,
    e1hosttandemalqxB1[[aux2]] = e1hosttandemalqxB1[[aux2]] + CALQUILADOB12[[ialq, 3]];];
  (* A nivel de las centrales de destino, Si la cabecera del destino no es la Tandem del departamento,
  y si dicha cabecera es distinta de la cabecera del origen,
  se suma para cada una de las cabeceras del destino el número de E1s de sus remotas identificadas como
  destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
  automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
  If[tandem1x[[aux1]] == 0 ∧ aux1 ≠ aux2,
    e1hosttandemalqxB1[[aux1]] = e1hosttandemalqxB1[[aux1]] + CALQUILADOB12[[ialq, 3]];];];
, {ialq, nalqB12}]

(* A nivel de las centrales de origen, si optó por la opción 3 o 4 (estimar costo de circuitos del rango A o B):
Si la cabecera del origen no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del destino,
se suma para cada una de las cabeceras del origen el número de E1s de sus remotas identificadas como
origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
Print[e1hosttandemalqxB1];

(* Este desarrollo esta identificando los E1s correspondientes a circuitos Intradepartamentales del tipo
B que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados se
estan acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxB1 *)

Print[Style["CIRCUITOS RANGO C", Bold, 16, Underlined, White, Background → Blue]];
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES",
  Bold, 14, Blue, Background → Yellow]];
OpenRead["input_C_AlquiladosC1.txt"];
(* Leyendo la información de los circuitos alquilados del rango C a nivel Interdepartamental:
La data viene así:
216    1
377    1
439    1
El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál),
el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado
con alguna de las centrales del modelo, en el ejemplo, los números 216 y 377 hacen
referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces,
en una fila aparecera el origen o destino y en otra el correspondiente destino u origen,
sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no
es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

Print["CALQUILADOC1"];
CALQUILADOC1 = Traducir[N[ReadList["input_C_AlquiladosC1.txt", Number, RecordLists → True]]];
Print[CALQUILADOC1];
(* Presenta la información del archivo de entrada de los circuitos alquilados
del rango C a nivel Interdepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [[origen o destino,número de E1s]] *)

Print["nalqC1"];
nalqC1 = Length[CALQUILADOC1];
Print[nalqC1];
(* Identifica el número de elementos de la lista CALQUILADOC1 *)

e1remotehostalqxC1 = zerox;
e1hosttandemalqxC1 = zerox;
TOTALE1ALQC1 = 0;
Print["TOTALE1ALQC1"];
Do[TOTALE1ALQC1 = TOTALE1ALQC1 +  $\frac{\text{CALQUILADOC1}[[\text{ialq}, 2]]}{2}$ ; , {ialq, nalqC1}];
Print[TOTALE1ALQC1];
(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo C que son atribuibles como
cargas en el tramo intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista),
se suman todos los E1s identificados (2do elemento de cada sublista). El
resultado se divide entre 2 porque cada circuito aparece dos veces *)

Print["e1remotehostalqxC1"];
Do[If[remote1x[[CALQUILADOC1[[ialq, 1]]]] == 1, e1remotehostalqxC1[[CALQUILADOC1[[ialq, 1]]]] =
  e1remotehostalqxC1[[CALQUILADOC1[[ialq, 1]]]] + CALQUILADOC1[[ialq, 2]]; , {ialq, nalqC1}]];
Print[e1remotehostalqxC1];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del
tipo C que estan asociados como punto de oriden o destino a cada central remota. Dentro
de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista),
identifica las centrales que son remotas y para cada una de ellas realiza
la sumatoria del total de E1s (2do elemento de cada sublista) *)

```



```

Print["e1hosttandemalqx1"];
Do[If[remote1x[[CALQUILADOC1[[ialq, 1]]] == 1, e1hosttandemalqx1[[allhostx[[CALQUILADOC1[[ialq, 1]]]]] =
    e1hosttandemalqx1[[allhostx[[CALQUILADOC1[[ialq, 1]]]]] + CALQUILADOC1[[ialq, 2]]];];
    If[remote1x[[CALQUILADOC1[[ialq, 1]]] == 0 & tandem1x[[CALQUILADOC1[[ialq, 1]]] == 0,
        e1hosttandemalqx1[[CALQUILADOC1[[ialq, 1]]] =
            e1hosttandemalqx1[[CALQUILADOC1[[ialq, 1]]] + CALQUILADOC1[[ialq, 2]]];];, {ialq, nalqC1}];
Print[e1hosttandemalqx1];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales
del tipo C que deberán ser añadidos a las cargas a nivel de centrales Host. Se
realiza lo siguiente: En la primera parte del código identifica las centrales
host que son cabeceras de las remotas en las cuales se identificó circuitos asociados,
y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas
(por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista
de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem
y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_Alquilados2C1.txt"];
(* Leyendo la información de los circuitos alquilados del rango C que son intradepartamentales:
La data viene así:
1    2    0
El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de E1s *)

Print["CALQUILADOC12"];
CALQUILADOC12 = Traducir[N[ReadList["input_C_Alquilados2C1.txt", Number, RecordLists -> True]]];
Print[CALQUILADOC12];
(* Presenta la información del archivo de entrada de los circuitos alquilados
del rango C a nivel Intradepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [ [origen,destino,número de E1s] ] *)

Print["nalqC12"];
nalqC12 = Length[CALQUILADOC12];
Print[nalqC12];
(* Identifica el número de elementos de la lista CALQUILADOC12 *)

Print["TOTALE1ALQC1"];
Do[TOTALE1ALQC1 = TOTALE1ALQC1 + CALQUILADOC12[[ialq, 3]];, {ialq, nalqC12}];
Print[TOTALE1ALQC1];
(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo C. Se suman todos los E1s
considerados como data (3er elemento de cada sublista). Nótese que dichos resultados se estan acumulando con la
estimación ya realizada anteriormente para la variable TOTALE1ALQC1 que correspondió a los E1s asociados a los
circuitos Interdepartamentales del tipo C que eran atribuibles como cargas en el tramo intradepartamental *)

Print["e1remotehostalqx1"];
Do[aux1 = CALQUILADOC12[[ialq, 1]];
    aux2 = CALQUILADOC12[[ialq, 2]];
    If[aux1 != aux2,
        If[remote1x[[aux1]] == 1, e1remotehostalqx1[[aux1]] = e1remotehostalqx1[[aux1]] + CALQUILADOC12[[ialq, 3]]];;
        If[remote1x[[aux2]] == 1, e1remotehostalqx1[[aux2]] = e1remotehostalqx1[[aux2]] + CALQUILADOC12[[ialq, 3]]];;
        , {ialq, nalqC12}];
Print[e1remotehostalqx1];
(* Esta identificando los E1s correspondientes a circuitos Intradepartamentales del
tipo C que estan asociados como punto de origen o destino a cada central remota. Se hace lo
siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que
son remotas y se suma para cada una de ellas el total de E1s (3er elemento de cada sublista). La segunda
parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se
suma para cada una de ellas el total de E1s (3er elemento de cada sublista). Nótese que dichos resultados
se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqx1 *)

Print["e1hosttandemalqx1"];
Do[aux1 = CALQUILADOC12[[ialq, 1]];
    aux2 = CALQUILADOC12[[ialq, 2]];
    If[aux1 != aux2,
        (* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
        If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]]];;
        If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]]];;
        (* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
en caso de no ser remota es claro que el nodo es identificado como Host *)
        If[tandem1x[[aux2]] == 0 & aux1 != aux2,
            e1hosttandemalqx1[[aux2]] = e1hosttandemalqx1[[aux2]] + CALQUILADOC12[[ialq, 3]]];;
        (* A nivel de las centrales de destino, Si la cabecera del destino no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del origen,
se suma para cada una de las cabeceras del destino el número de E1s de sus remotas identificadas como
destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
        If[tandem1x[[aux1]] == 0 & aux1 != aux2,
            e1hosttandemalqx1[[aux1]] = e1hosttandemalqx1[[aux1]] + CALQUILADOC12[[ialq, 3]]];;];

```



```

, {ialq, nalqC12}}];
(* A nivel de las centrales de origen, si optó por la opción 5 (estimar costo de circuitos del rango C):
Si la cabecera del origen no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del destino,
se suma para cada una de las cabeceras del origen el número de EIs de sus remotas identificadas como
origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,
automáticamente se imputa a dicha central los EIs del circuito (3er elemento la sublista) *)
Print[e1hosttandemalqxC1];
(* Este desarrollo esta identificando los EIs correspondientes a circuitos Intradepartamentales del tipo
C que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados se
estan acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxC1 *)

Print[Style["CIRCUITOS LOCAL LIMA", Bold, 16, Underlined, White, Background -> Blue]];
Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_Alquilados2LL.txt"];
(* Leyendo la información de los circuitos alquilados Local Lima que son intradepartamentales:
La data viene así:
1 2 0
El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de EIs *)

Print["CALQUILADOLL2"];
CALQUILADOLL2 = Traducir[N[ReadList["input_C_Alquilados2LL.txt", Number, RecordLists -> True]]];
Print[CALQUILADOLL2];
(* Presenta la información del archivo de entrada de los circuitos alquilados
Local Lima a nivel Intradepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [ [origen,destino,número de EIs] ] *)

Print["nalqLL2"];
nalqLL2 = Length[CALQUILADOLL2];
Print[nalqLL2];
(* Identifica el número de elementos de la lista CALQUILADOLL2 *)

Print["TOTALE1ALQLL"];
TOTALE1ALQLL = 0; Do[TOTALE1ALQLL = TOTALE1ALQLL + CALQUILADOLL2[[ialq, 3]], {ialq, nalqLL2}];
Print[TOTALE1ALQLL];
(* Identifica el total de EIs asociados a los circuitos Interdepartamentales Local Lima. Se suman todos los EIs
considerados como data (3er elemento de cada sublista). Nótese que dichos resultados se estan acumulando con la
estimación ya realizada anteriormente para la variable TOTALE1ALQLL que correspondió a los EIs asociados a los
circuitos Interdepartamentales Local Lima que eran atribuibles como cargas en el tramo intradepartamental *)

Print["e1remotehostalqxLL"];
e1remotehostalqxLL = zerox;
Do[aux1 = CALQUILADOLL2[[ialq, 1]];
aux2 = CALQUILADOLL2[[ialq, 2]];
If[aux1 != aux2,
If[remote1x[[aux1]] == 1, e1remotehostalqxLL[[aux1]] = e1remotehostalqxLL[[aux1]] + CALQUILADOLL2[[ialq, 3]];];
If[remote1x[[aux2]] == 1, e1remotehostalqxLL[[aux2]] = e1remotehostalqxLL[[aux2]] + CALQUILADOLL2[[ialq, 3]];];];
, {ialq, nalqLL2}];
Print[e1remotehostalqxLL];
(* Esta identificando los EIs correspondientes a circuitos Intradepartamentales Local
Lima que estan asociados como punto de origen o destino a cada central remota. Se hace lo
siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que
son remotas y se suma para cada una de ellas el total de EIs (3er elemento de cada sublista). La segunda
parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se
suma para cada una de ellas el total de EIs (3er elemento de cada sublista). Nótese que dichos resultados
se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqxLL *)

Print["e1hosttandemalqxLL"];
e1hosttandemalqxLL = zerox;
Do[aux1 = CALQUILADOLL2[[ialq, 1]];
aux2 = CALQUILADOLL2[[ialq, 2]];
If[aux1 != aux2,
(* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]];];
If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]];];
(* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
en caso de no ser remota es claro que el nodo es identificado como Host *)
If[tandem1x[[aux2]] == 0 & aux1 != aux2,
e1hosttandemalqxLL[[aux2]] = e1hosttandemalqxLL[[aux2]] + CALQUILADOLL2[[ialq, 3]];];
(* A nivel de las centrales de destino, Si la cabecera del destino no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del origen,
se suma para cada una de las cabeceras del destino el número de EIs de sus remotas identificadas como
destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
automáticamente se imputa a dicha central los EIs del circuito (3er elemento la sublista) *)
If[tandem1x[[aux1]] == 0 & aux1 != aux2,
e1hosttandemalqxLL[[aux1]] = e1hosttandemalqxLL[[aux1]] + CALQUILADOLL2[[ialq, 3]];];];
, {ialq, nalqLL2}];

```

```

(* A nivel de las centrales de origen, si optó por la opción (estimar costo de circuitos Local Lima):
  Si la cabecera del origen no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del destino,
se suma para cada una de las cabeceras del origen el número de EIs de sus remotas identificadas como
  origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,
automáticamente se imputa a dicha central los EIs del circuito (3er elemento la sublista) *)
Print[e1hosttandemalqxLL];
(* Este desarrollo esta identificando los EIs correspondientes a circuitos Intradepartamentales Local
  Lima que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados se
  estan acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxLL *)

Print[Style["CIRCUITOS LOCAL PROVINCIA", Bold, 16, Underlined, White, Background -> Blue]],
Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background -> Yellow]];
OpenRead["input_C_Alquilados2LP.txt"];
(* Leyendo la información de los circuitos alquilados Local Provincia que son intradepartamentales:
  La data viene así:
  1    2    0
  El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de EIs *)

Print["CALQUILADOLP2"];
CALQUILADOLP2 = Traducir[N[ReadList["input_C_Alquilados2LP.txt", Number, RecordLists -> True]]];
Print[CALQUILADOLP2];
(* Presenta la información del archivo de entrada de los circuitos alquilados
  Local Provincia a nivel Intradepartamental como una lista donde cada elemento es a su
  vez una lista que presenta los datos de cada fila: [ [origen,destino,número de EIs] ] *)

Print["nalqLP2"];
nalqLP2 = Length[CALQUILADOLP2];
Print[nalqLP2];
(* Identifica el número de elementos de la lista CALQUILADOLP2 *)

Print["TOTALE1ALQLP"];
TOTALE1ALQLP = 0; Do[TOTALE1ALQLP = TOTALE1ALQLP + CALQUILADOLP2[[ialq, 3]], {ialq, nalqLP2}];
Print[TOTALE1ALQLP];
(* Identifica el total de EIs asociados a los circuitos Interdepartamentales Local Provincia. Se
  suman todos los EIs considerados como data (3er elemento de cada sublista). Nótese que dichos
  resultados se estan acumulando con la estimación ya realizada anteriormente para la variable
  TOTALE1ALQLP que correspondió a los EIs asociados a los circuitos Interdepartamentales
  Local Provincia que eran atribuibles como cargas en el tramo intradepartamental *)

Print["e1remotehostalqxLP"];
e1remotehostalqxLP = zerox;
Do[aux1 = CALQUILADOLP2[[ialq, 1]];
  aux2 = CALQUILADOLP2[[ialq, 2]];
  If[aux1 != aux2,
    If[remote1x[[aux1]] == 1, e1remotehostalqxLP[[aux1]] = e1remotehostalqxLP[[aux1]] + CALQUILADOLP2[[ialq, 3]];];
    If[remote1x[[aux2]] == 1, e1remotehostalqxLP[[aux2]] = e1remotehostalqxLP[[aux2]] + CALQUILADOLP2[[ialq, 3]];];];
  , {ialq, nalqLP2}];
Print[e1remotehostalqxLP];
(* Esta identificando los EIs correspondientes a circuitos Intradepartamentales Local
  Provincia que estan asociados como punto de oriden o destino a cada central remota. Se hace lo
  siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que
  son remotas y se suma para cada una de ellas el total de EIs (3er elemento de cada sublista). La segunda
  parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se
  suma para cada una de ellas el total de EIs (3er elemento de cada sublista). Nótese que dichos resultados
  se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqxLP *)

Print["e1hosttandemalqxLP"];
e1hosttandemalqxLP = zerox;
Do[aux1 = CALQUILADOLP2[[ialq, 1]];
  aux2 = CALQUILADOLP2[[ialq, 2]];
  If[aux1 != aux2,
    (* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
    If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]];];
    If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]];];
    (* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
    en caso de no ser remota es claro que el nodo es identificado como Host *)
    If[tandem1x[[aux2]] == 0 & aux1 != aux2,
      e1hosttandemalqxLP[[aux2]] = e1hosttandemalqxLP[[aux2]] + CALQUILADOLP2[[ialq, 3]];];
    (* A nivel de las centrales de destino, Si la cabecera del destino no es la Tandem del departamento,
    y si dicha cabecera es distinta de la cabecera del origen,
    se suma para cada una de las cabeceras del destino el número de EIs de sus remotas identificadas como
    destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
    automáticamente se imputa a dicha central los EIs del circuito (3er elemento la sublista) *)
    If[tandem1x[[aux1]] == 0 & aux1 != aux2,
      e1hosttandemalqxLP[[aux1]] = e1hosttandemalqxLP[[aux1]] + CALQUILADOLP2[[ialq, 3]];];];
  , {ialq, nalqLP2}];

```

```

(* A nivel de las centrales de origen, si optó por la opción (estimar costo de circuitos Local Provincia):
Si la cabecera del origen no es la Tandem del departamento,
y si dicha cabecera es distinta de la cabecera del destino,
se suma para cada una de las cabeceras del origen el número de E1s de sus remotas identificadas como
origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
Print[e1hosttandemalqxLP];
(* Este desarrollo esta identificando los E1s correspondientes a circuitos Intradepartamentales Local Provincia
que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados se
están acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxLP *)

Print[Style["CIRCUITOS RANGO A SATELITAL", Bold, 16, Underlined, White, Background → Blue]];
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES",
Bold, 14, Blue, Background → Yellow]];

OpenRead["input_C_AlquiladosAS.txt"];
(* Leyendo la información de los circuitos alquilados del rango A Satelital a nivel Interdepartamental:
La data viene así:
39      1
377     1
377     1
El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál),
el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado
con alguna de las centrales del modelo, en el ejemplo, los números 39 y 377 hacen
referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces,
en una fila aparecera el origen o destino y en otra el correspondiente destino u origen,
sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no
es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

Print["CALQUILADOAS"];
CALQUILADOAS = Traducir[N[ReadList["input_C_AlquiladosAS.txt", Number, RecordLists → True]]];
Print[CALQUILADOAS];
(* Presenta la información del archivo de entrada de los circuitos alquilados del
rango A Satelital a nivel Interdepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [ [origen o destino ,número de E1s] ] *)

Print["nalqAS"];
nalqAS = Length[CALQUILADOAS];
Print[nalqAS];
(* Identifica el número de elementos de la lista CALQUILADOAS *)

e1remotehostalqxAS = zerox;
e1hosttandemalqxAS = zerox;
TOTALE1ALQAS = 0;
Print["TOTALE1ALQAS"];
Do[TOTALE1ALQAS = TOTALE1ALQAS +  $\frac{\text{CALQUILADOAS}[[\text{ialq}, 2]]}{2}$ ; , {ialq, nalqAS}];
Print[TOTALE1ALQAS];
(* Identifica el total de E1s asociados a los circuitos
Interdepartamentales del tipo A Satelital que son atribuibles como cargas en el tramo
intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista),
se suman todos los E1s identificados (2do elemento de cada sublista). El
resultado se divide entre 2 porque cada circuito aparece dos veces *)

Print["e1remotehostalqxAS"];
Do[If[remote1x[CALQUILADOAS[[ialq, 1]]] == 1, e1remotehostalqxAS[[CALQUILADOAS[[ialq, 1]]] =
e1remotehostalqxAS[[CALQUILADOAS[[ialq, 1]]] + CALQUILADOAS[[ialq, 2]]]; , {ialq, nalqAS}];
Print[e1remotehostalqxAS];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo
A Satelital que estan asociados como punto de origen o destino a cada central remota. Dentro
de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista),
identifica las centrales que son remotas y para cada una de ellas realiza
la sumatoria del total de E1s (2do elemento de cada sublista) *)

Print["e1hosttandemalqxAS"];
Do[If[remote1x[CALQUILADOAS[[ialq, 1]]] == 1, e1hosttandemalqxAS[[allhostx[CALQUILADOAS[[ialq, 1]]]] =
e1hosttandemalqxAS[[allhostx[CALQUILADOAS[[ialq, 1]]]] + CALQUILADOAS[[ialq, 2]]]; ,
If[remote1x[CALQUILADOAS[[ialq, 1]]] == 0 ^ tandem1x[CALQUILADOAS[[ialq, 1]]] == 0,
e1hosttandemalqxAS[[CALQUILADOAS[[ialq, 1]]] =
e1hosttandemalqxAS[[CALQUILADOAS[[ialq, 1]]] + CALQUILADOAS[[ialq, 2]]]; , {ialq, nalqAS}];
Print[e1hosttandemalqxAS];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales
del tipo A Satelital que deberán ser añadidos a las cargas a nivel de centrales
Host. Se realiza lo siguiente: En la primera parte del código identifica las centrales
host que son cabeceras de las remotas en las cuales se identificó circuitos asociados,
y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas
(por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista
de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem

```

y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print[Style["CIRCUITOS RANGO B SATELITAL", Bold, 16, Underlined, White, Background → Blue]];
```

```
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES", Bold, 14, Blue, Background → Yellow]];
```

```
OpenRead["input_C_AlquiladosBS.txt"];
```

(* Leyendo la información de los circuitos alquilados del rango B Satelital a nivel Interdepartamental:

La data viene así:

```
39    1
```

```
377   1
```

```
377   1
```

El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál), el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado con alguna de las centrales del modelo, en el ejemplo, los números 39 y 377 hacen referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces, en una fila aparecera el origen o destino y en otra el correspondiente destino u origen, sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

```
Print["CALQUILADOBS"];
```

```
CALQUILADOBS = Traducir[N[ReadList["input_C_AlquiladosBS.txt", Number, RecordLists → True]]];
```

```
Print[CALQUILADOBS];
```

(* Presenta la información del archivo de entrada de los circuitos alquilados del rango B Satelital a nivel Interdepartamental como una lista donde cada elemento es a su vez una lista que presenta los datos de cada fila: [[origen o destino ,número de E1s]] *)

```
Print["nalqBS"];
```

```
nalqBS = Length[CALQUILADOBS];
```

```
Print[nalqBS];
```

(* Identifica el número de elementos de la lista CALQUILADOBS *)

```
e1remotehostalqxBS = zerox;
```

```
e1hosttandemalqxBS = zerox;
```

```
TOTALE1ALQBS = 0;
```

```
Print["TOTALE1ALQBS"];
```

```
Do[TOTALE1ALQBS = TOTALE1ALQBS +  $\frac{\text{CALQUILADOBS}[[\text{ialq}, 2]]}{2}$ ;;, {ialq, nalqBS}];
```

```
Print[TOTALE1ALQBS];
```

(* Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo B Satelital que son atribuibles como cargas en el tramo intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista), se suman todos los E1s identificados (2do elemento de cada sublista). El resultado se divide entre 2 porque cada circuito aparece dos veces *)

```
Print["e1remotehostalqxBS"];
```

```
Do[If[remote1x[[CALQUILADOBS[[ialq, 1]]] == 1, e1remotehostalqxBS[[CALQUILADOBS[[ialq, 1]]] = e1remotehostalqxBS[[CALQUILADOBS[[ialq, 1]]] + CALQUILADOBS[[ialq, 2]]];;, {ialq, nalqBS}];
```

```
Print[e1remotehostalqxBS];
```

(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo B Satelital que estan asociados como punto de oriden o destino a cada central remota. Dentro de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista), identifica las centrales que son remotas y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print["e1hosttandemalqxBS"];
```

```
Do[If[remote1x[[CALQUILADOBS[[ialq, 1]]] == 1, e1hosttandemalqxBS[[allhostx[[CALQUILADOBS[[ialq, 1]]]]] = e1hosttandemalqxBS[[allhostx[[CALQUILADOBS[[ialq, 1]]]]] + CALQUILADOBS[[ialq, 2]]];;
```

```
If[remote1x[[CALQUILADOBS[[ialq, 1]]] == 0 & tandem1x[[CALQUILADOBS[[ialq, 1]]] == 0,
```

```
e1hosttandemalqxBS[[CALQUILADOBS[[ialq, 1]]] =
```

```
e1hosttandemalqxBS[[CALQUILADOBS[[ialq, 1]]] + CALQUILADOBS[[ialq, 2]]];;, {ialq, nalqBS}];
```

```
Print[e1hosttandemalqxBS];
```

(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo B Satelital que deberán ser añadidos a las cargas a nivel de centrales Host. Se realiza lo siguiente: En la primera parte del código identifica las centrales host que son cabeceras de las remotas en las cuales se identificó circuitos asociados, y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas (por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print[Style["CIRCUITOS RANGO C SATELITAL", Bold, 16, Underlined, White, Background → Blue]];
```

```
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES", Bold, 14, Blue, Background → Yellow]];
```

```
OpenRead["input_C_AlquiladosCS.txt"];
```

(* Leyendo la información de los circuitos alquilados del rango C Satelital a nivel Interdepartamental:

La data viene así:

```
39    1
```



```
377 1
377 1
```

El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál), el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado con alguna de las centrales del modelo, en el ejemplo, los números 39 y 377 hacen referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces, en una fila aparecera el origen o destino y en otra el correspondiente destino u origen, sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)

```
Print["CALQUILADOCS"];
CALQUILADOCS = Traducir[N[ReadList["input_C_AlquiladosCS.txt", Number, RecordLists -> True]]];
Print[CALQUILADOCS];
(* Presenta la información del archivo de entrada de los circuitos alquilados del
rango C Satelital a nivel Interdepartamental como una lista donde cada elemento es a su vez
una lista que presenta los datos de cada fila: [ [origen o destino ,número de E1s] ] *)
```

```
Print["nalqCS"];
nalqCS = Length[CALQUILADOCS];
Print[nalqCS];
(* Identifica el número de elementos de la lista CALQUILADOCS *)
```

```
e1remotehostalqxCS = zerox;
e1hosttandemalqxCS = zerox;
TOTALE1ALQCS = 0;
Print["TOTALE1ALQCS"];
Do[TOTALE1ALQCS = TOTALE1ALQCS +  $\frac{\text{CALQUILADOCS}[[\text{ialq}, 2]]}{2}$ ;;, {ialq, nalqCS}];
Print[TOTALE1ALQCS];
```

(*) Identifica el total de E1s asociados a los circuitos Interdepartamentales del tipo C Satelital que son atribuibles como cargas en el tramo intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista), se suman todos los E1s identificados (2do elemento de cada sublista). El resultado se divide entre 2 porque cada circuito aparece dos veces *)

```
Print["e1remotehostalqxCS"];
Do[If[remote1x[[CALQUILADOCS[[ialq, 1]]]] == 1, e1remotehostalqxCS[[CALQUILADOCS[[ialq, 1]]]] =
e1remotehostalqxCS[[CALQUILADOCS[[ialq, 1]]]] + CALQUILADOCS[[ialq, 2]];;, {ialq, nalqCS}];
Print[e1remotehostalqxCS];
(* Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo
C Satelital que estan asociados como punto de oriden o destino a cada central remota. Dentro
de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista),
identifica las centrales que son remotas y para cada una de ellas realiza
la sumatoria del total de E1s (2do elemento de cada sublista) *)
```

```
Print["e1hosttandemalqxCS"];
Do[If[remote1x[[CALQUILADOCS[[ialq, 1]]]] == 1, e1hosttandemalqxCS[[allhostx[[CALQUILADOCS[[ialq, 1]]]]]] =
e1hosttandemalqxCS[[allhostx[[CALQUILADOCS[[ialq, 1]]]]]] + CALQUILADOCS[[ialq, 2]];;, {ialq, nalqCS}];
If[remote1x[[CALQUILADOCS[[ialq, 1]]]] == 0 & tandem1x[[CALQUILADOCS[[ialq, 1]]]] == 0,
e1hosttandemalqxCS[[CALQUILADOCS[[ialq, 1]]]] =
e1hosttandemalqxCS[[CALQUILADOCS[[ialq, 1]]]] + CALQUILADOCS[[ialq, 2]];;, {ialq, nalqCS}];
Print[e1hosttandemalqxCS];
```

(*) Esta identificando los E1s correspondientes a circuitos interdepartamentales del tipo C Satelital que deberán ser añadidos a las cargas a nivel de centrales Host. Se realiza lo siguiente: En la primera parte del código identifica las centrales host que son cabeceras de las remotas en las cuales se identificó circuitos asociados, y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas (por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras que no son Tandem y para cada una de ellas realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

```
Print[Style["CIRCUITOS NO OPERADORES", Bold, 16, Underlined, White, Background -> Blue]];
Print[Style["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES",
Bold, 14, Blue, Background -> Yellow]];
```

```
OpenRead["input_C_AlquiladosABCnop.txt"];
(* Leyendo la información de los circuitos alquilados
de los no operadores que son Interdepartamentales. La data viene así:
39 8
39 7
216 9
```

El primer dato especifica uno de los puntos del circuito (origen o destino, no se sabe cuál), el segundo dato especifica la capacidad en E1s. Cada origen o destino esta identificado con alguna de las centrales del modelo, en el ejemplo, los números 39 y 216 hacen referencia a las centrales identificadas con dichos números. Cada circuito aparece 2 veces, en una fila aparecera el origen o destino y en otra el correspondiente destino u origen, sin embargo la información no esta ordenada (las filas no son consecutivas), razón por la cual no es posible identificar a priori cuales son los pares ordenados asociados a un mismo circuito *)


```

Print["CALQUILADOABCnop"];
CALQUILADOABCnop = Ceiling[Traducir[N[ReadList["input_C_AlquiladosABCnop.txt", Number, RecordLists → True]]]];
Print[CALQUILADOABCnop];
(* Presenta la información del archivo de entrada de los circuitos alquilados de
los no operadores a nivel Interdepartamental como una lista donde cada elemento es a su
vez una lista que presenta los datos de cada fila: [[origen o destino, número de EIs]] *)

Print["nalqABCnop"];
nalqABCnop = Length[CALQUILADOABCnop];
Print[nalqABCnop];
(* Identifica el número de elementos de la lista CALQUILADOABCnop *)

e1remotehostalqxABCnop = zerox;
e1hosttandemalqxABCnop = zerox;
TOTALE1ALQABCnop = 0;
Print["TOTALE1ALQABCnop"];

Do[TOTALE1ALQABCnop = TOTALE1ALQABCnop +  $\frac{\text{CALQUILADOABCnop}[[\text{ialq}, 2]]}{2}$ ;, {ialq, nalqABCnop}];
Print[TOTALE1ALQABCnop];
(* Identifica el total de EIs asociados a los circuitos alquilados
Interdepartamentales de los no operadores que son atribuibles como cargas en el tramo
intradepartamental. Independientemente del nodo de referencia (1er dato de cada sublista),
se suman todos los EIs identificados (2do elemento de cada sublista). El
resultado se divide entre 2 porque cada circuito aparece dos veces *)

Print["e1remotehostalqxABCnop"];
Do[If[remote1x[[CALQUILADOABCnop[[ialq, 1]]] == 1, e1remotehostalqxABCnop[[CALQUILADOABCnop[[ialq, 1]]] =
e1remotehostalqxABCnop[[CALQUILADOABCnop[[ialq, 1]]] + CALQUILADOABCnop[[ialq, 2]]];, {ialq, nalqABCnop}];
Print[e1remotehostalqxABCnop];
(* Esta identificando los EIs correspondientes a circuitos alquilados interdepartamentales
de los no operadores que estan asociados como punto de origen o destino a cada central
remota. Dentro de la lista de nodos relacionados con dichos circuitos (1er dato de cada sublista),
identifica las centrales que son remotas y para cada una de ellas realiza
la sumatoria del total de EIs (2do elemento de cada sublista) *)

Print["e1hosttandemalqxABCnop"];
Do[If[remote1x[[CALQUILADOABCnop[[ialq, 1]]] == 1, e1hosttandemalqxABCnop[[allhostx[[CALQUILADOABCnop[[ialq, 1]]]] =
e1hosttandemalqxABCnop[[allhostx[[CALQUILADOABCnop[[ialq, 1]]]] + CALQUILADOABCnop[[ialq, 2]]];
If[remote1x[[CALQUILADOABCnop[[ialq, 1]]] == 0 ^ tandem1x[[CALQUILADOABCnop[[ialq, 1]]] == 0,
e1hosttandemalqxABCnop[[CALQUILADOABCnop[[ialq, 1]]] =
e1hosttandemalqxABCnop[[CALQUILADOABCnop[[ialq, 1]]] + CALQUILADOABCnop[[ialq, 2]]];, {ialq, nalqABCnop}];
Print[e1hosttandemalqxABCnop];
(* Esta identificando los EIs correspondientes a circuitos alquilados interdepartamentales de los no operadores
que deberán ser añadidos a las cargas a nivel de centrales Host. En la primera parte del código identifica
las centrales host que son cabeceras de las remotas en las cuales se identificó circuitos asociados,
y para cada una de dichas cabeceras realiza la sumatoria de los circuitos asociados a sus remotas
(por jerarquía de la red). En la segunda parte del código se identifica directamente dentro de la lista
de nodos relacionados (1er dato de cada sublista) las centrales que son cabeceras y que no son Tandem
y para cada una de ellas se realiza la sumatoria del total de EIs (2do elemento de cada sublista) *)

Print[Style["CIRCUITOS INTRADEPARTAMENTALES", Bold, 14, Blue, Background → Yellow]];
OpenRead["input_C_Alquilados2ABCnop.txt"];
(* Leyendo la información de los circuitos alquilados
de los no operadores que son intradepartamentales: La data viene así:
377    261    3
183    190    2
39     76     2
El primer dato corresponde a la central de origen,
el segundo dato a la central de destino y el tercer dato al número de EIs *)

Print["CALQUILADOABC2nop"];
CALQUILADOABC2nop = Ceiling[Traducir[N[ReadList["input_C_Alquilados2ABCnop.txt", Number, RecordLists → True]]]];
Print[CALQUILADOABC2nop];
(* Presenta la información del archivo de entrada de los circuitos alquilados de
los no operadores nivel Intradepartamental como una lista donde cada elemento es a su
vez una lista que presenta los datos de cada fila: [ [origen,destino,número de EIs] ] *)

Print["nalqABC2nop"];
nalqABC2nop = Length[CALQUILADOABC2nop];
Print[nalqABC2nop];
(* Identifica el número de elementos de la lista CALQUILADOABC2nop *)

Print["TOTALE1ALQABCnop"];
Do[TOTALE1ALQABCnop = TOTALE1ALQABCnop + CALQUILADOABC2nop[[ialqnop, 3]];, {ialqnop, nalqABC2nop}];
Print[TOTALE1ALQABCnop];
(* Identifica el total de EIs asociados a los circuitos alquilados Interdepartamentales de los no
operadores. Se suman todos los EIs considerados como data (3er elemento de cada sublista). Nótese que
dichos resultados se estan acumulando con la estimación ya realizada anteriormente para la variable
TOTALE1ALQB2 que correspondió a los EIs asociados a los circuitos alquilados Interdepartamentales

```

de los no operadores que eran atribuibles como cargas en el tramo intradepartamental *)

```
Print["e1remotehostalqxABCnop"];
```

```
Do[aux1 = CALQUILADOABC2nop[[ialq, 1]];
```

```
aux2 = CALQUILADOABC2nop[[ialq, 2]];
```

```
If[aux1 ≠ aux2,
```

```
If[remote1x[[aux1]] == 1,
```

```
e1remotehostalqxABCnop[[aux1]] = e1remotehostalqxABCnop[[aux1]] + CALQUILADOABC2nop[[ialq, 3]]];];
```

```
If[remote1x[[aux2]] == 1, e1remotehostalqxABCnop[[aux2]] =
```

```
e1remotehostalqxABCnop[[aux2]] + CALQUILADOABC2nop[[ialq, 3]]];];];
```

```
, {ialq, nalqABC2nop}];
```

```
Print[e1remotehostalqxABCnop];
```

(* Esta identificando los E1s correspondientes a circuitos alquilados Intradepartamentales

de los no operadores que estan asociados como punto de oriden o destino a cada central

remota. Si se optó por estimar los costos de los circutos del tipo B se hace lo

siguiente: La primera parte del código identifica todos los nodos de origen (1er elemento de cada sublista) que

son remotas y se suma para cada una de ellas el total de E1s (3er elemento de cada sublista). La segunda

parte del código identifica todos los nodos de destino (2do elemento de cada sublista) que son remotas y se

suma para cada una de ellas el total de E1s (3er elemento de cada sublista). Nótese que dichos resultados

se estan acumulando con la estimación ya realizada anteriormente para la variable e1remotehostalqxABCnop *)

```
Print["e1hosttandemalqxABCnop"];
```

```
Do[aux1 = CALQUILADOABC2nop[[ialq, 1]];
```

```
aux2 = CALQUILADOABC2nop[[ialq, 2]];
```

```
If[aux1 ≠ aux2,
```

```
(* El código identifica el origen y el destino de cada circuito (1er y 2do elemento de cada sublista) *)
```

```
If[remote1x[[aux1]] == 1, aux1 = allhostx[[aux1]]];];
```

```
If[remote1x[[aux2]] == 1, aux2 = allhostx[[aux2]]];];
```

```
(* Si alguno de dichos nodos de origen y destino es remota el código identifica su respectivas cabeceras,
```

```
en caso de no ser remota es claro que el nodo es identificado como Host *)
```

```
If[tandem1x[[aux2]] == 0 ∧ aux1 ≠ aux2,
```

```
e1hosttandemalqxABCnop[[aux2]] = e1hosttandemalqxABCnop[[aux2]] + CALQUILADOABC2nop[[ialq, 3]]];];
```

```
(* A nivel de las centrales de destino, si la cabecera del destino no es la Tandem del departamento,
```

```
y si dicha cabecera es distinta de la cabecera del origen,
```

```
se suma para cada una de las cabeceras del destino el número de E1s de sus remotas identificadas como
```

```
destino de los circuitos (3er elemento de cada sublista). Si el destino no es remota, osea si es Host,
```

```
automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)
```

```
If[tandem1x[[aux1]] == 0 ∧ aux1 ≠ aux2, e1hosttandemalqxABCnop[[aux1]] =
```

```
e1hosttandemalqxABCnop[[aux1]] + CALQUILADOABC2nop[[ialq, 3]]];];];
```

```
, {ialq, nalqABC2nop}];
```

(* A nivel de las centrales de origen, si la cabecera del origen no es la Tandem del departamento,

y si dicha cabecera es distinta de la cabecera del destino,

se suma para cada una de las cabeceras del origen el número de E1s de sus remotas identificadas como

origen de los circuitos (3er elemento de cada sublista). Si el origen no es remota, osea si es Host,

automáticamente se imputa a dicha central los E1s del circuito (3er elemento la sublista) *)

```
Print[e1hosttandemalqxABCnop];
```

(* Este desarrollo esta identificando los E1s correspondientes a circuitos alquilados Intradepartamentales de

los no operadores que deberán ser añadidos a las cargas a nivel de centrales Host. Nótese que dichos resultados

se estan acumulando con la estimación ya realizada anteriormente para la variable e1hosttandemalqxABCnop *)

```
PRINT["ESTIMACIÓN DE CARGAS EN E1S ASOCIADAS A LOS CIRCUITOS DE  
INTERCONEXIÓN Y QUE SON IMPUTABLES A LAS REDES INTRADEPARTAMENTALES"]
```

```
PRINT["CIRCUITOS POR ENLACES DE INTERCONEXIÓN"]
```

```
OpenRead["input_C_ITX.txt"];
```

```
(* Lee la información correspondiente a los circuitos de interconexión. La información se presenta de la  
siguiente manera:
```

```
439 39  
439 40  
39 2  
132 1  
223 1
```

```
El primer dato especifica el PDI de la red local de Tdp donde se realiza la interconexión, y el segundo dato  
corresponde al número de E1s del enlace *)
```

```
Print["CdeITX"];
```

```
CdeITX=Traducir[N[ReadList["input_C_ITX.txt",Number,RecordLists→True]]];
```

```
Print[CdeITX];
```

```
(* Presenta la información de los circuitos de interconexión como una lista donde cada elemento es a su vez una  
lista que presenta los datos de cada fila: [[PDI,número de E1s]] *)
```

```
Print["ncdeitx"];
```

```
ncdeitx=Length[CdeITX];
```

```
Print[ncdeitx];
```

```
(* Identifica el número de elementos de la lista CdeITX *)
```

```
e1remotehostitxx=zeror;
```

```
e1hosttandemitxx=zeror;
```

```
TOTALE1ITX=0;
```

```
TOTALE1ITXaux=0;
```

```
CITX=zeror;
```

```
Print["TOTALE1ITX"];
```

```
Do[TOTALE1ITX=TOTALE1ITX+CdeITX[[ialq,2]],{ialq,ncdeitx}];
```

```
Print[TOTALE1ITX];
```

```
(* Sumatoria del total de E1s de Interconexión existentes, equivale a la sumatoria de los 2dos elementos de cada  
una de las sublistas *)
```

```
Print["TOTALE1ITXaux"]
```

```
Do[If[tandem1x[[CdeITX[[ialq,1]]]]==0,TOTALE1ITXaux=TOTALE1ITXaux+CdeITX[[ialq,2]]];,{ialq,ncdeitx}];
```

```
Print[TOTALE1ITXaux]
```

```
(* Sumatoria del total de E1s de Interconexión existentes sólo para de los casos en los cuales el PDI  
especificado no es una central Tandem. Para dichos caso equivale a la sumatoria de los 2dos elementos de cada una  
de las sublistas *)
```

```
Print["CITX"]
```

```
Do[CITX[[CdeITX[[ialq,1]]]]]=CITX[[CdeITX[[ialq,1]]]]+CdeITX[[ialq,2]],{ialq,ncdeitx}];
```

```
Print[CITX]
```

```
(* Identifica el total de E1s de interconexión que existen en cada central (obviamente habrá sólo 1 por  
departamento). Para ello identifica a que departamento pertenece cada PDI y luego totaliza el número de E1s que  
existe en cada departamento *)
```

```
Print["e1remotehostitxx"];
```

```
Do[If[remote1x[[CdeITX[[ialq,1]]]]==1,
```

```
    e1remotehostitxx[[CdeITX[[ialq,1]]]] =
```

```
    e1remotehostitxx[[CdeITX[[ialq,1]]]]+CdeITX[[ialq,2]],{ialq,ncdeitx}];
```

```
Print[e1remotehostitxx];
```

```
(* Identifica el total de E1s a nivel R-H que estan asociados o vinculados con los enlaces de interconexión. Para  
ello, identifica los PDIs que son remotas y para cada uno de dichos casos totaliza el total de E1s de  
interconexión que se encuentran vinculados con dichos nodos *)
```

```
Print["e1hosttandemitxx"];
```

```
Do[If[remote1x[[CdeITX[[ialq,1]]]]==1,
```

```
    e1hosttandemitxx[[allhostx[[CdeITX[[ialq,1]]]]]] = e1hosttandemitxx[[allhostx[[CdeITX[[ialq,1]]]]]] + CdeITX[[ialq,2]];]
```

```
    If[remote1x[[CdeITX[[ialq,1]]]]==0 ^ tandem1x[[CdeITX[[ialq,1]]]]==0,
```

```
        e1hosttandemitxx[[CdeITX[[ialq,1]]]] = e1hosttandemitxx[[CdeITX[[ialq,1]]]] + CdeITX[[ialq,2]],{ialq,ncdeitx}];
```

```
Print[e1hosttandemitxx];
```

```
(* Identifica el total de E1s a nivel H-T que estan asociados o vinculados con los enlaces de interconexión. En  
la primera parte del código identifica las centrales host que son cabeceras de las remotas en las cuales se  
identificó circuitos de interconexión, y para cada una de dichas cabeceras realiza la sumatoria de los circuitos  
asociados a sus remotas (por jerarquía de la red). En la segunda parte del código se identifica directamente  
dentro de la lista de PDIs (1er dato de cada sublista) las centrales que son cabeceras y que no son Tandem y para  
cada una de ellas se realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)
```

ALGORITMO DE PRIM

```

Print[Style["ALGORITMO DE PRIM", Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

(***** en estrella con fibra enterrada *****)
e1remotehostalqxABC2 = zerox;
e1hosttandemalqxABC2 = zerox;
If[ConsideraPRIM ≠ 0,
  nodo = zerox;
  nodonoradio = zerox;
  numeronodos = 0;
  numeronodosnoradio = 0;
  prime1remotehostx = zerox; (* a nivel de voz *)
  primloadremotehostLocx = zerox; (* Voz Local *)
  primloadremotehostLDNx = zerox; (* Voz LDN *)
  primloadremotehostLDIx = zerox; (* Voz LDI *)
  prime1remotehostalqxA1 = zerox; (* circuitos tipo A *)
  prime1remotehostalqxB1 = zerox; (* circuitos tipo B *)
  prime1remotehostalqxC1 = zerox; (* circuitos tipo C *)
  prime1remotehostalqxLL = zerox; (* circuitos tipo Local LIMA *)
  prime1remotehostalqxLP = zerox; (* circuitos tipo Local Provincias *)
  prime1remotehostalqxAS = zerox; (* circuitos tipo A Satelital*)
  prime1remotehostalqxBS = zerox; (* circuitos tipo B Satelital*)
  prime1remotehostalqxCS = zerox; (* circuitos tipo C Satelital*)
  prime1remotehostalqxABC2 = zerox; (* otros circuitos tipo A, B y C*)
  prime1remotehostalqxABCnop = zerox; (* circuitos de No Operadores*)
  prime1remotehostitxx = zerox; (* circuitos de Interconexión *)
  prime1remotehostadslx = zerox; (* Los de ADSL *)
  Do[If[host1x[[iax]] == 1,
    numeronodos = 1;
    numeronodosnoradio = 1;
    nodo[[numeronodos]] = iax;
    nodonoradio[[numeronodosnoradio]] = iax;,
    If[esAnillopx[[iax]] == 0 ∧ MemberQ[{Enterrado, Aereo, Urbana, Radio},
      transtechx[[iax]]], numeronodos++;
    nodo[[numeronodos]] = iax;];
    (*Inicio de Cambios en el Algoritmo de PRIM para que sea similar al del modelo en access*)
    If[esAnillopx[[iax]] == 0 ∧ MemberQ[{Enterrado, Aereo, Urbana},
      transtechx[[iax]]],
      numeronodosnoradio++;
      nodonoradio[[numeronodosnoradio]] = iax;];]; (**)
  If[host1x[[iax + 1]] == 1 ∧ numeronodos > 2, (* comienzo Prim *)
    Print["Prim"];
    (*Todos los nodos*)
    disttotal = 0;
    nodoconectado = Table[0, {ix, numeronodos}];
    nodoconectado[[1]] = 1;
    (* vale 1 si ya forma parte del grafo *)
    quienmeconecto = zerox;
    quienmeconecto[[1]] = 1;
    nodoencontrado = 1;
    While[nodoencontrado ≠ 0,
      distaux = ∞;
      nodoencontrado = 0;
      nodoalqueseconecta = 0;
      For[iinod = 1, iinod < numeronodos,
        For[jjnod = iinod + 1, jjnod < numeronodos + 1, If[(distance[nodo[[iinod]], nodo[[jjnod]]] < distaux ∨
          nodoencontrado == 0) ∧ ((nodoconectado[[iinod]] == 1 ∧ nodoconectado[[jjnod]] == 0) ∨
            (nodoconectado[[jjnod]] == 1 ∧ nodoconectado[[iinod]] == 0)),
          distaux = distance[nodo[[iinod]], nodo[[jjnod]]];
          nodoencontrado = If[nodoconectado[[iinod]] == 0, iinod, jjnod];
          nodoalqueseconecta = If[nodoconectado[[jjnod]] == 0, iinod, jjnod];]; jjnod++; iinod++;
      If[nodoencontrado ≠ 0, nodoconectado[[nodoencontrado]] = 1;
      quienmeconecto[[nodoencontrado]] = nodoalqueseconecta;];
      disttotal = disttotal +  $\frac{\text{distaux}}{\text{factorNoLinealNoUrban}}$ ;];
    (*Nodos de Fibra*)
    disttotalnoradio = 0;
    nodoconectadonoradio = Table[0, {ix, numeronodosnoradio}];
    nodoconectadonoradio[[1]] = 1;
    (* vale 1 si ya forma parte del grafo *)
    quienmeconectonoradio = zerox;
    quienmeconectonoradio[[1]] = 1;
    nodoencontradonoradio = 1;

```

```

numeronodonormal = Table[0, {ix, numeronodosnoradio}];
For[ibx = 1, ibx ≤ numeronodosnoradio,
  For[icx = 1, icx ≤ numeronodos, If[nodonoradio[[ibx]] == nodo[[icx]], numeronodonormal[[ibx]] = icx];
  icx++];
ibx++];
While[nodoencontradonoradio ≠ 0,
  distauxnoradio = ∞;
  nodoencontradonoradio = 0;
  nodoalqueseconectanoradio = 0;
  For[iinod = 1, iinod < numeronodosnoradio,
    For[jjnod = iinod + 1, jjnod < numeronodosnoradio + 1,
      If[(distance[nodonoradio[[iinod]], nodonoradio[[jjnod]] < distauxnoradio ∨ nodoencontradonoradio == 0) ∧
        ((nodoconectadonoradio[[iinod]] == 1 ∧ nodoconectadonoradio[[jjnod]] == 0) ∨ (nodoconectadonoradio[[jjnod]] == 1 ∧
          nodoconectadonoradio[[iinod]] == 0)), distauxnoradio = distance[nodonoradio[[iinod]], nodonoradio[[jjnod]]];
      nodoencontradonoradio = If[nodoconectadonoradio[[iinod]] == 0, iinod, jjnod];
      nodoalqueseconectanoradio = If[nodoconectadonoradio[[jjnod]] == 0, iinod, jjnod];]; jjnod++; iinod++;
  If[nodoencontradonoradio ≠ 0, nodoconectadonoradio[nodoencontradonoradio] = 1;
  aquienmeconectonoradio[nodoencontradonoradio] = nodoalqueseconectanoradio];
  disttotalnoradio = disttotalnoradio +  $\frac{\text{distauxnoradio}}{\text{factorNoLinealNoUrban}}$ ;];
For[ibx = 2, ibx ≤ numeronodosnoradio,
  aquienmeconecto[numeronodonormal[[ibx]]] = numeronodonormal[aquienmeconectonoradio[[ibx]]]; ibx++];

For[ibx = 2, ibx ≤ numeronodos, distaux = distance[nodo[[ibx]], nodo[aquienmeconecto[[ibx]]]];
  disthostx[nodo[[ibx]]] =  $\frac{\text{distaux}}{\text{factorNoLinealx[nodo[[ibx]]]}}$ ;
  repfibrax[nodo[[ibx]]] =  $\frac{\text{distaux}}{\text{factorNoLinealx[nodo[[ibx]]]}}$ ;
  Conexion[nodo[[ibx]]] = nodo[aquienmeconecto[[ibx]]];
  If[(ibx == 1) ∧ (ConsideraPRIM == 2) ∧ (transtechx[nodo[[ibx]]] == Enterrado),
    disthostx[nodo[[ibx]]] = 2 * disthostx[nodo[[ibx]]];
    repfibrax[nodo[[ibx]]] = (*2*) repfibrax[nodo[[ibx]]];];
  (*Se le retira el producto por 2*)
  If[(ConsideraPRIM == 3) ∧ (transtechx[nodo[[ibx]]] == Enterrado),
    disthostx[nodo[[ibx]]] = 2 * disthostx[nodo[[ibx]]];
    repfibrax[nodo[[ibx]]] = (*2*) repfibrax[nodo[[ibx]]];];
  (*Se le retira el producto por 2*)
  disttotal = disttotal +  $\frac{\text{distaux}}{\text{factorNoLinealNoUrban}}$ ;
  ibx++];
(* ahora actualizar numero de e1s *)
Do[prime1remotehostx[[ix]] = e1remotehostx[[ix]];
  primloadremotehostLocx[[ix]] = loadremotehostLocx[[ix]];
  primloadremotehostLDNx[[ix]] = loadremotehostLDNx[[ix]];
  primloadremotehostLDIx[[ix]] = loadremotehostLDIx[[ix]];
  prime1remotehostalqxA1[[ix]] = e1remotehostalqxA1[[ix]];
  prime1remotehostalqxB1[[ix]] = e1remotehostalqxB1[[ix]];
  prime1remotehostalqxC1[[ix]] = e1remotehostalqxC1[[ix]];
  prime1remotehostalqxLL[[ix]] = e1remotehostalqxLL[[ix]];
prime1remotehostalqxLP[[ix]] = e1remotehostalqxLP[[ix]];
prime1remotehostalqxAS[[ix]] = e1remotehostalqxAS[[ix]];
prime1remotehostalqxBs[[ix]] = e1remotehostalqxBs[[ix]];
prime1remotehostalqxCs[[ix]] = e1remotehostalqxCs[[ix]];
  prime1remotehostalqxABc2[[ix]] = e1remotehostalqxABc2[[ix]];
  prime1remotehostalqxABcnoP[[ix]] = e1remotehostalqxABcnoP[[ix]];
  prime1remotehostitxx[[ix]] = e1remotehostitxx[[ix]];
  prime1remotehostadslx[[ix]] = e1remotehostadslx[[ix]]; {ix, nx}];
aquienmeconecto2 = aquienmeconecto;
For[iinod = 2, iinod < numeronodos + 1, While[
  (aquienmeconecto2[[iinod]] ≠ 0) ∧ (aquienmeconecto2[[iinod]] ≠ 1), e1remotehostx[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostx[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostx[nodo[[iinod]]];
  loadremotehostLocx[nodo[aquienmeconecto2[[iinod]]]] =
  loadremotehostLocx[nodo[aquienmeconecto2[[iinod]]]] + primloadremotehostLocx[nodo[[iinod]]];
  loadremotehostLDNx[nodo[aquienmeconecto2[[iinod]]]] =
  loadremotehostLDNx[nodo[aquienmeconecto2[[iinod]]]] + primloadremotehostLDNx[nodo[[iinod]]];
  loadremotehostLDIx[nodo[aquienmeconecto2[[iinod]]]] =
  loadremotehostLDIx[nodo[aquienmeconecto2[[iinod]]]] + primloadremotehostLDIx[nodo[[iinod]]];
  e1remotehostalqxA1[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostalqxA1[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostalqxA1[nodo[[iinod]]];
  e1remotehostalqxB1[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostalqxB1[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostalqxB1[nodo[[iinod]]];
  e1remotehostalqxC1[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostalqxC1[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostalqxC1[nodo[[iinod]]];
  e1remotehostalqxLL[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostalqxLL[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostalqxLL[nodo[[iinod]]];
  e1remotehostalqxLP[nodo[aquienmeconecto2[[iinod]]]] =
  e1remotehostalqxLP[nodo[aquienmeconecto2[[iinod]]]] + prime1remotehostalqxLP[nodo[[iinod]]];

```



```

e1remotehostalqxAS[[nodo[[aquiemeconecto2[[iinod]]]]] =
  e1remotehostalqxAS[[nodo[[aquiemeconecto2[[iinod]]]]] + prime1remotehostalqxAS[[nodo[[iinod]]]];
e1remotehostalqxBs[[nodo[[aquiemeconecto2[[iinod]]]]] =
  e1remotehostalqxBs[[nodo[[aquiemeconecto2[[iinod]]]]] + prime1remotehostalqxBs[[nodo[[iinod]]]];
e1remotehostalqxCS[[nodo[[aquiemeconecto2[[iinod]]]]] =
  e1remotehostalqxCS[[nodo[[aquiemeconecto2[[iinod]]]]] + prime1remotehostalqxCS[[nodo[[iinod]]]];
e1remotehostalqxABC2[[nodo[[aquiemeconecto2[[iinod]]]]] =
  e1remotehostalqxABC2[[nodo[[aquiemeconecto2[[iinod]]]]] + prime1remotehostalqxABC2[[nodo[[iinod]]]];
e1remotehostalqxABCnop[[nodo[[aquiemeconecto2[[iinod]]]]] =
  e1remotehostalqxABCnop[[nodo[[aquiemeconecto2[[iinod]]]]] + prime1remotehostalqxABCnop[[nodo[[iinod]]]];
e1remotehostadslx[[nodo[[aquiemeconecto2[[iinod]]]]] = e1remotehostadslx[[nodo[[aquiemeconecto2[[iinod]]]]] +
  prime1remotehostadslx[[nodo[[iinod]]]];
e1remotehostitxx[[nodo[[aquiemeconecto2[[iinod]]]]] = e1remotehostitxx[[nodo[[aquiemeconecto2[[iinod]]]]] +
  prime1remotehostitxx[[nodo[[iinod]]]];
aquiemeconecto2[[iinod]] = aquiemeconecto[[aquiemeconecto2[[iinod]]]];];
iinod++;];
, {iax, nx - 1}];
];

(*****Fin Algoritmo de Prim para conectar las remotas*****)
Tcs = {"", "Enlaces de Itx", "LD en Rango A", "LD en Rango B", "LD en Rango C"};
Print["Valores Luego de PRIM"];
loadremotehostx = loadremotehostLocx + loadremotehostLDNx + loadremotehostLDIx;
Print[TableForm[
  Table[{ix, idx[[ix]], e1remotehostx[[ix]], loadremotehostLocx[[ix]], loadremotehostLDNx[[ix]], loadremotehostLDIx[[ix]],
    loadremotehostx[[ix]], e1remotehostalqxAl[[ix]], e1remotehostalqxB1[[ix]], e1remotehostalqxC1[[ix]],
    e1remotehostalqxLL[[ix]], e1remotehostalqxLP[[ix]], e1remotehostalqxAS[[ix]], e1remotehostalqxBs[[ix]],
    e1remotehostalqxCS[[ix]], e1remotehostalqxABC2[[ix]], e1remotehostalqxABCnop[[ix]],
    e1remotehostadslx[[ix]], e1remotehostitxx[[ix]]}, {ix, nx}], TableDirections -> {Column, Row},
  TableHeadings -> {None, {"ix", "idx", "E1s Voz", "Erlangs Local", "Erlangs LDN", "Erlangs LDI",
    "Erlangs Total", "E1s Rango A", "E1s Rango B", "E1s Rango C", "E1s Local Lima", "E1s Local Prov",
    "E1s Rango A Sat", "E1s Rango B Sat", "E1s Rango C Sat", "E1s Otros", "E1s No Op", "E1s ADSL", "E1s ITX"}}]];

```

ACUMULACION DE E1s A NIVEL DE ANILLOS

```
PRINT["ACUMULADO DEL NÚMERO DE E1s A NIVEL DE ANILLOS"];
```

```
PRINT["ACUMULADO PROVINCIALES"];
```

```

e1Anillop = zerop;      (* Voz *)
loadAnilloLocp = zerop; (* Voz Local *)
loadAnilloLDNp = zerop; (* Voz LDN *)
loadAnilloLDIp = zerop; (* Voz LDI *)
e1AnilloalqpA1 = zerop; (* Circuitos Rango A *)
e1AnilloalqpB1 = zerop; (* Circuitos Rango B *)
e1AnilloalqpC1 = zerop; (* Circuitos Rango C *)
e1AnilloalqpLL = zerop; (* Circuitos Local Lima *)
e1AnilloalqpLP = zerop; (* Circuitos Local Provincias *)
e1AnilloalqpAS = zerop; (* Circuitos Rango A Satelital*)
e1AnilloalqpBS = zerop; (* Circuitos Rango B Satelital*)
e1AnilloalqpCS = zerop; (* Circuitos Rango C Satelital*)
e1AnilloalqpABC2 = zerop; (* Otros Circuitos Rango A, B y C*)
e1AnilloalqpABCnop = zerop; (* Circuitos de No Operadores*)
e1Anilloadslp = zerop;   (* ADSL *)
e1Anilloitxp = zerop;    (* Circuitos de Interconexión *)
ProvAnillo = 1;
esAnillo = 0;

Do[
  If[host1x[[ix]] == 1,
    If[esAnillopx[[ix]] == 1,
      ProvAnillo = provx[[ix]];
      esAnillo = 1;,
      esAnillo = 0;
    ];
  ];
If[esAnillo == 1,
  e1Anillop[[ProvAnillo]] =
    e1Anillop[[ProvAnillo]] + If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostx[[ix]] * If[host1x[[ix]] == 1, 0, 1];
  loadAnilloLocp[[ProvAnillo]] = loadAnilloLocp[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * loadremotehostLocx[[ix]] * If[host1x[[ix]] == 1, 0, 1];
  loadAnilloLDNp[[ProvAnillo]] = loadAnilloLDNp[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * loadremotehostLDNx[[ix]] * If[host1x[[ix]] == 1, 0, 1];
  loadAnilloLDIp[[ProvAnillo]] = loadAnilloLDIp[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * loadremotehostLDIx[[ix]] * If[host1x[[ix]] == 1, 0, 1];
  e1AnilloalqpA1[[ProvAnillo]] = e1AnilloalqpA1[[ProvAnillo]] +

```

```

    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalqxA1[[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpB1[[ProvAnillo]] = e1AnilloalqpB1[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalqxB1[[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpC1[[ProvAnillo]] = e1AnilloalqpC1[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalqxC1[[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpLL[[ProvAnillo]] = e1AnilloalqpLL[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalqxLL[[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpLP[[ProvAnillo]] = e1AnilloalqpLP[[ProvAnillo]] + If[esAnillopx[[ix]] == 0, 0, 1] *
    e1remotehostalq(LP)[ix]] * If[host1x[[ix]] == 1, 0, 1]; e1AnilloalqpAS[[ProvAnillo]] =
    e1AnilloalqpAS[[ProvAnillo]] + If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalq(xAS)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpBS[[ProvAnillo]] = e1AnilloalqpBS[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalq(xBS)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpCS[[ProvAnillo]] = e1AnilloalqpCS[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalq(xCS)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpABC2[[ProvAnillo]] = e1AnilloalqpABC2[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalq(xABC2)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1AnilloalqpABCnop[[ProvAnillo]] = e1AnilloalqpABCnop[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostalq(xABCnop)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1Anilloadslp[[ProvAnillo]] = e1Anilloadslp[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostadsl(x)[ix]] * If[host1x[[ix]] == 1, 0, 1];
e1Anilloitxp[[ProvAnillo]] = e1Anilloitxp[[ProvAnillo]] +
    If[esAnillopx[[ix]] == 0, 0, 1] * e1remotehostit(x)(x)[ix]] * If[host1x[[ix]] == 1, 0, 1];
];
, {ix, nx}];
loadAnillop = loadAnilloLocp + loadAnilloLDNp + loadAnilloLDIp;

```

(* Se estima el número de EIs a nivel de cada anillo provincia para cada servicio. Para ello, se identifica cada provincia (igual a la provincia de cada cabecera) y dentro de cada provincia donde efectivamente existe un anillo provincial se suma de todas las remotas que forman parte del anillo la respectiva variable e1remotehostx, loadremotehostLocx, loadremotehostLDNx, loadremotehostLDIx, loadremotehostx, e1remotehostalqx(A1), e1remotehostalqxB1, e1remotehostalqxC1, e1remotehostalqxLL, e1remotehostalq(LP), e1remotehostalq(xAS), e1remotehostalq(xBS), e1remotehostalq(xCS), e1remotehostalq(xABC2), e1remotehostalq(xABCnop), e1remotehostadsl(x), e1remotehostit(x)(x) *)

```

Print[TableForm[Table[{ip, provTXTp[[ip]], e1Anillop[[ip]], loadAnilloLocp[[ip]], loadAnilloLDNp[[ip]],
    loadAnilloLDIp[[ip]], e1AnilloalqpA1[[ip]], e1AnilloalqpB1[[ip]], e1AnilloalqpC1[[ip]], e1AnilloalqpLL[[ip]],
    e1AnilloalqpLP[[ip]], e1AnilloalqpAS[[ip]], e1AnilloalqpBS[[ip]], e1AnilloalqpCS[[ip]], e1AnilloalqpABC2[[ip]],
    e1AnilloalqpABCnop[[ip]], e1Anilloadslp[[ip]], e1Anilloitxp[[ip]]}, {ip, np}], TableSpacing -> {0, 1},
    TableDirections -> {Column, Row}, TableHeadings -> {None, {"ip", "Nombre Provincia", "e1Anillop",
    "loadAnilloLocp", "loadAnilloLDNp", "loadAnilloLDIp", "e1AnilloalqpA1", "e1AnilloalqpB1",
    "e1AnilloalqpC1", "e1AnilloalqpLL", "e1AnilloalqpLP", "e1AnilloalqpAS", "e1AnilloalqpBS",
    "e1AnilloalqpCS", "e1AnilloalqpABC2", "e1AnilloalqpABCnop", "e1Anilloadslp", "e1Anilloitxp"}}]];

```

PRINCIPIO DE ACUMULADO DEPARTAMENTALES

```

e1Anillod = zerod;      (* Voz *)
loadAnilloLocd = zerod; (* Voz Local *)
loadAnilloLDNd = zerod; (* Voz LDN *)
loadAnilloLDId = zerod; (* Voz LDI *)
e1AnilloalqdA1 = zerod; (* Circuitos Rango A *)
e1AnilloalqdB1 = zerod; (* Circuitos Rango B *)
e1AnilloalqdC1 = zerod; (* Circuitos Rango C *)
e1AnilloalqdLL = zerod; (* Circuitos Local Lima *)
e1AnilloalqdLP = zerod; (* Circuitos Local Provincia *)
e1AnilloalqdAS = zerod; (* Circuitos Rango A Satelital*)
e1AnilloalqdBS = zerod; (* Circuitos Rango B Satelital*)
e1AnilloalqdCS = zerod; (* Circuitos Rango C Satelital*)
e1AnilloalqdABC2 = zerod; (* Otros Circuitos Rango A, B y C *)
e1AnilloalqdABCnop = zerod; (* Circuitos de No Operadores *)
e1Anilloadsld = zerod;  (* ADSL *)
e1AnilloRDd = zerod;    (* Rutas directas *)
e1Anilloitxd = zerod;    (* Circuitos de Interconexión *)
DepAnillo = 0;
esAnillo = 0;

```

```

Do[If[AnilloCabeceras[[ip]] != 0,
    If[tandem1x[[AnilloCabeceras[[ip]]]] == 1,
        If[esAnillodx[[AnilloCabeceras[[ip]]]] != NA,
            DepAnillo = deptx[[AnilloCabeceras[[ip]]]];
            esAnillo = 1;
            esAnillo = 0;
        ];
    ];
If[esAnillo == 1, e1Anillod[[DepAnillo]] = e1Anillod[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
    e1hosttandemx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
loadAnilloLocd[[DepAnillo]] = loadAnilloLocd[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
    loadhosttandemLocx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
loadAnilloLDNd[[DepAnillo]] = loadAnilloLDNd[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
    loadhosttandemLDNx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];

```

```

loadAnilloLDId[[DepAnillo]] = loadAnilloLDId[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  loadhosttandemLDIx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdA1[[DepAnillo]] = e1AnilloalqdA1[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalqxA1[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdB1[[DepAnillo]] = e1AnilloalqdB1[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalqxB1[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdC1[[DepAnillo]] = e1AnilloalqdC1[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalqxC1[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdLL[[DepAnillo]] = e1AnilloalqdLL[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalqxLL[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdLP[[DepAnillo]] = e1AnilloalqdLP[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq Lup[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdAS[[DepAnillo]] = e1AnilloalqdAS[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq xAS[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdBs[[DepAnillo]] = e1AnilloalqdBs[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq xBS[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdCS[[DepAnillo]] = e1AnilloalqdCS[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq xCS[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdABC2[[DepAnillo]] = e1AnilloalqdABC2[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq xABC2[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloalqdABCnop[[DepAnillo]] = e1AnilloalqdABCnop[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemalq xABCnop[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1Anilloadsld[[DepAnillo]] = e1Anilloadsld[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemadslx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1Anilloitxd[[DepAnillo]] = e1Anilloitxd[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1hosttandemitxx[[AnilloCabeceras[[ip]]]] * If[tandem1x[[AnilloCabeceras[[ip]]]] == 1, 0, 1];
e1AnilloRDd[[DepAnillo]] = e1AnilloRDd[[DepAnillo]] + If[esAnillodx[[AnilloCabeceras[[ip]]]] == NA, 0, 1] *
  e1RDx[[AnilloCabeceras[[ip]]]];
];
];
, {ip, np}];
loadAnillod = loadAnilloLocd + loadAnilloLDNd + loadAnilloLDId;

```

(* Se estima el número de E1s de cada anillo departamental para cada servicio. Para ello, se identifica cada departamento (igual al departamento de cada Tandem) y dentro de cada departamento donde efectivamente existe un anillo departamental se suma de todas las cabeceras que forman parte del anillo la respectiva variable e1hosttandemx, loadhosttandemLocx, loadhosttandemLDNx, loadhosttandemLDIx, loadhosttandemx, e1hosttandemalqxA1, e1hosttandemalqxB1, e1hosttandemalqxC1, e1hosttandemalqxLL, e1hosttandemalq Lup, e1hosttandemalq xAS, e1hosttandemalq xBS, e1hosttandemalq xCS, e1hosttandemalq xABC2, e1hosttandemalq xABCnop, e1hosttandemadslx, e1hosttandemitxx, e1RDx *)

```

Print[TableForm[Table[{id, deptTXTd[[id]], e1Anillod[[id]], loadAnilloLocd[[id]], loadAnilloLDNd[[id]],
  loadAnilloLDId[[id]], e1AnilloalqdA1[[id]], e1AnilloalqdB1[[id]], e1AnilloalqdC1[[id]], e1AnilloalqdLL[[id]],
  e1AnilloalqdLP[[id]], e1AnilloalqdAS[[id]], e1AnilloalqdBs[[id]], e1AnilloalqdCS[[id]], e1AnilloalqdABC2[[id]],
  e1AnilloalqdABCnop[[id]], e1Anilloadsld[[id]], e1Anilloitxd[[id]]}, {id, nd}}, TableSpacing -> {0, 1},
  TableDirections -> {Column, Row}, TableHeadings -> {None, {"id", "Nombre Departamento", "e1Anillod",
  "loadAnilloLocd", "loadAnilloLDNd", "loadAnilloLDId", "e1AnilloalqdA1", "e1AnilloalqdB1",
  "e1AnilloalqdC1", "e1AnilloalqdLL", "e1AnilloalqdLP", "e1AnilloalqdAS", "e1AnilloalqdBs",
  "e1AnilloalqdCS", "e1AnilloalqdABC2", "e1AnilloalqdABCnop", "e1Anilloadsld", "e1Anilloitxd"}}]];

```

PRINT["A NIVEL DE TODOS LOS SERVICIOS"]

```

e1remotehostTOTALx = Ceiling[e1remotehostx + e1remotehostadslx + e1remotehostalqxA1 + e1remotehostalqxB1 +
  e1remotehostalqxC1 + e1remotehostalqxLL + e1remotehostalq Lup + e1remotehostalq xAS + e1remotehostalq xBS +
  e1remotehostalq xCS + e1remotehostalq xABC2 + e1remotehostalq xABCnop + e1remotehostitxx];
e1hosttandemTOTALx = Ceiling[e1hosttandemx + e1hosttandemadslx + e1hosttandemalqxA1 + e1hosttandemalqxB1 +
  e1hosttandemalqxC1 + e1hosttandemalqxLL + e1hosttandemalq Lup + e1hosttandemalq xAS + e1hosttandemalq xBS +
  e1hosttandemalq xCS + e1hosttandemalq xABC2 + e1hosttandemalq xABCnop + e1hosttandemitxx];
(* Estima el total de E1s a nivel RH y HT, incluyendo todos los servicios: Cargas de Voz,
Cargas de circuitos alquilados de los 3 rangos, Cargas de otros circuitos,
Adsl, Cargas de circuitos de no Operadores y Enlaces de Interconexión. *)
Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], e1remotehostTOTALx[[ix]], e1hosttandemTOTALx[[ix]]}, {ix, nx}},
  TableSpacing -> {0, 1}, TableDirections -> {Column, Row},
  TableHeadings -> {None, {"ix", "idx", "Nombre Central", "e1remotehostTOTALx", "e1hosttandemTOTALx"}}]];

e1AnilloTOTALp = Ceiling[
  e1Anillop + e1AnilloalqpA1 + e1AnilloalqpB1 + e1AnilloalqpC1 + e1AnilloalqpLL + e1AnilloalqpLP + e1AnilloalqpAS +
  e1AnilloalqpBS + e1AnilloalqpCS + e1AnilloalqpABC2 + e1AnilloalqpABCnop + e1Anilloadslp + e1Anilloitxp];
(* Estima el total de E1s de los anillos provinciales para todos los servicios. Para
ellos suma los E1s asociados a cada uno de los servicios. *)
Print[TableForm[Table[{ip, provTXTp[[ip]], e1AnilloTOTALp[[ip]]}, {ip, np}}, TableSpacing -> {0, 1},
  TableDirections -> {Column, Row}, TableHeadings -> {None, {"ip", "Nombre Provincia", "e1AnilloTOTALp"}}]];

e1AnilloTOTALd = Ceiling[
  e1Anillod + e1AnilloalqdA1 + e1AnilloalqdB1 + e1AnilloalqdC1 + e1AnilloalqdLL + e1AnilloalqdLP + e1AnilloalqdAS +
  e1AnilloalqdBs + e1AnilloalqdCS + e1AnilloalqdABC2 + e1AnilloalqdABCnop + e1Anilloadsld + e1Anilloitxd];

```

72

Modelo_Integral_ACK.nb

	<pre>(* Estima el total de EIs de los anillos departamentales para todos los servicios. Para ellos suma los EIs asociados a cada uno de los servicios. *) Print[TableForm[Table[{id, deptTXTd[[id]], e1AnilloTOTALd[[id]]}, {id, nd}], TableSpacing -> {0, 1}, TableDirections -> {Column, Row}, TableHeadings -> {None, {"id", "Nombre Departamento", "e1AnilloTOTALd"}}]]; </pre>	
--	---	--

CALCULO DE COSTOS EN CONMUTACIÓN

Dimensionamiento, Costeo, Distribución de Costo e Identificación de Costo Atribuible a Tráfico para Centrales de Conmutación y Unidades Remotas

Lectura de Datos

■ Centrales

■ Estructura del Archivo de Centrales

El archivo de centrales tiene la siguiente estructura:

- ID Central
- Departamento
- Provincia
- Nombre Central
- Funcion
- Lineas PSTN
- Lineas RSDI Basicas
- Lineas RSDI Primarias
- Numero de Remotas (Asociadas a Central Cabecera)

Dicho archivo tiene formato de texto separado por comas e incluye los nombres de los campos.

■ Lectura del Archivo de Centrales {CenComInput, CenComN }

```
( * lee el archivo incluyendo cabecera *)
CenComAux = Import["ModCom_ListadoCentrales.txt", "CSV"];

(*extrae cabecera *)
CenComInput = Rest[CenComAux];

(*numero de elementos *)
CenComN = Length[CenComInput];

```

■ Parametros

■ Precios {CenComPrec}

```
( * lee el archivo incluyendo cabecera *)
CenComAux = ReadList["ModCom_PreciarioCentrales.txt", String, RecordLists -> True];
(*subdivide cada linea por el separador ";"*)
CenComAux = StringSplit[CenComAux[[1]], ";"];

(*extrae cabecera *)
CenComPrec = Rest[CenComAux];
(* convierte valores de cadena a numeros*)
m = Length[CenComPrec];
Do[CenComPrec[[i, 3]] = ToExpression[CenComPrec[[i, 3]]]; (* ITEM *)
  CenComPrec[[i, 5]] = ToExpression[CenComPrec[[i, 5]]]; (* PRECIO UNITARIO *)
  , {i, 1, m}];

```

■ Factores {CenComFactores, CenComPorcInst}

```
( * lee el archivo incluyendo cabecera *)
CenComAux = Import["ModCom_Factores.txt", "CSV"];

(*extrae cabecera *)
CenComFactores = Rest[CenComAux];

CenComPorcInst = CenComFactores[[1, 3]];

```

Calculo de Parametros de Dimensionamiento de las Centrales

■ Enlaces (E0' s) {CenComE0s}

```
CenComE0s = Table[0, {i, 1, CenComN}];
Do[idCentralMFija = CenComInput[[i, 1]];
  idPosCentralMIntegral = idinvertx[[idCentralMFija]];
  CenComE0s[[i]] = trunkx[[idPosCentralMIntegral]], {i, 1, CenComN}];
```

■ Remotas por Cabecera {CenComRemCab}

```
(*Crea un array para contener el id_central y el numero de remotas que atiende*)
CenComRemCab = Table[{CenComInput[[i, 1]], CenComInput[[i, 9]]}, {i, 1, CenComN}];
```

■ Acondicionamiento de Archivo de E0s

```
Module[
  {aux},
  (* Actualiza numero de E0's *)
  (* san isidro neax *)
  CenComE0s[[400]] = 105;
  (* tandemlimax *)
  CenComE0s[[252]] = 21 287;
];
```

Dimensionamiento de Centrales

■ Funciones Auxiliares

■ BuscarV

Funcion que busca el valor indicado en una matriz y devuelve el elemento que corresponde de la columna solicitada (El primer valor debe ser el valor de Cero)
Es una función propia de MS Excel.

```
BuscarV[mValorBuscado_, mMatrizBuscarEn_, mColumna_] := Module[{i, j, k, a, m, AuxV},
  m = Length[mMatrizBuscarEn];
  AuxV = mMatrizBuscarEn[[m, mColumna]];
  Do[
    If[mValorBuscado <= mMatrizBuscarEn[[i, 1]],
      AuxV = mMatrizBuscarEn[[i, mColumna]];
      Return[AuxV];
    ];
  , {i, 1, m}];
  Return[AuxV];
];
```

■ RedondearMas

Funcion que calcula el mayor numero entero, alejándose del Cero (Equivale a la funcion Ceiling para valores positivos y a Floor para valores negativos)
Es una función propia de MS Excel

```
RedondearMas[x_] := If[x ≥ 0, Ceiling[x], Floor[x]];
```

■ RedondearMenos

Funcion que calcula el menor numero entero , acercándose al Cero (Equivale a la funcion Floor para valores positivos y a Ceiling para valores negativos)
Es una función propia de MS Excel

```
RedondearMenos[x_] := If[x >= 0, Floor[x], Ceiling[x]];
```


■ **Matriz de Categorías de Costos {CenComCatCosto, - CenComCatCostoAccesoPSTN, CenComCatCostoAccesoBAS, CenComCatCostoAccesoPRI, CenComCatCostoTrafico, CenComCatCostoSenalizacion - }**

Se define una matriz que contiene las categorías de costo que serán utilizadas para distribuir el costo de los elementos de las centrales (Cabeceras y Unidades Remotas) posteriormente. Por simplificación se ha considerado conveniente utilizar el mismo vector tanto para Cabeceras como para Unidades Remotas. En el caso de categorías no utilizadas, simplemente no tendrán valor asignado.

```
CenComCatCosto = {  
  "ACCESO-PSTN", (* 1 *)  
  "ACCESO-RDSI BASICO", (* 2 *)  
  "ACCESO-RDSI PRIMARIO", (* 3 *)  
  "TRAFICO", (* 4 *)  
  "SEÑALIZACION" (* 5 *)  
};
```

Se definen adicionalmente un conjunto de símbolos que actuarán como "índices" de la matriz definida anteriormente para hacer referencia al tipo de costo deseado.

```
CenComCatCostoAccesoPSTN = 1;  
CenComCatCostoAccesoBAS = 2;  
CenComCatCostoAccesoPRI = 3;  
CenComCatCostoTrafico = 4;  
CenComCatCostoSenalizacion = 5;
```

■ **Matrices de Elementos de cada Subsistema de una Central Cabecera**

Cada Central Cabecera se ha dividido en diversos subsistemas, cada uno de los cuales contiene elementos o "equipos" (armarios, tarjetas, almacenes, cables, etc.) Se ha desarrollado un patrón para facilitar el trabajo con estos elementos. Dicho patrón es el siguiente:

Para cada Subsistema:

- Definición de Matrices generales
 - Matriz de Nombre de los elementos (Títulos)
 - Matriz de Precio de los elementos
 - Matriz de Matriz de distribución de costo de los elementos
 - Matriz auxiliar para la definción de elementos. (Equivale a la definición de un tipo/estructura de datos)

- Para cada elemento del subsistema
 - Comentario conteniendo el nombre del elemento
 - Variable auxiliar para indexar el elemento al interior de las matrices del subsistema
 - Asignación del nombre del elemento
 - Asignación del precio del elemento

- Para las Centrales de Conmutación
 - Definición de Matrices Generales
 - Matriz que contiene la estructura de la central, agrupados en subsistemas y elementos: auxElem. (Equivale a la definición de un tipo/estructura de datos)
 - Matriz que contiene el vector de precios de la central, agrupados en subsistemas y elementos: auxPrec
 - Matriz que contiene el nombre de los elementos de la central, agrupados en subsistemas y elementos: auxTit
 - Matriz que contiene la distribución de los elementos de la central, para cada tipo de costo, agrupados en subsistemas y elementos: auxDist (Equivale a la definición de un tipo/estructura de datos)

■ Matriz de Elementos del Subsistema de Abonado {CenComTituloSA, CenComPreciosSA, CenComDistribucionSA, auxSA}

```
CenComTituloSA = Table["", {i, 1, 14}];
CenComPreciosSA = Table[0, {i, 1, 14}];
CenComDistribucionSA = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 14}];
auxSA = Table[0, {i, 1, 14}];

(* TARJETAS DE LINEA *)
a = 1;
CenComTituloSA[[a]] = "TARJETA PARA 8 ABONADOS ACA - 3";
CenComPreciosSA[[a]] = CenComPrec[[9, 5]];

a = 2;
CenComTituloSA[[a]] = "TARJETA KR PARA LSM DE ACA - 3";
CenComPreciosSA[[a]] = CenComPrec[[10, 5]];

(* LSM's *)
a = 3;
CenComTituloSA[[a]] = "LSM ACA - 3 PARA 128 AB.POU.";
CenComPreciosSA[[a]] = CenComPrec[[8, 5]];

a = 4;
CenComTituloSA[[a]] = "LSM PARA 64 ACC 2 B + D INCL.60";
CenComPreciosSA[[a]] = CenComPrec[[6, 5]];

a = 5;
CenComTituloSA[[a]] = "NUEVO LSM - PRA CON ACC.30 B + D";
CenComPreciosSA[[a]] = CenComPrec[[7, 5]];

(* ALMACENES GENERALES *)
a = 6;
CenComTituloSA[[a]] = "ALMACEN RPBC (DUPLICADO)";
CenComPreciosSA[[a]] = CenComPrec[[2, 5]];

a = 7;
CenComTituloSA[[a]] = "ALMACEN SE - PRM (50) 8 AB.GEN.";
CenComPreciosSA[[a]] = CenComPrec[[4, 5]];

(* ELEMENTOS DE CONEXION CON GSS Y UNIDADES REMOTAS *)
a = 8;
CenComTituloSA[[a]] = "TARJETA JTC PARA RDSI";
CenComPreciosSA[[a]] = CenComPrec[[5, 5]];

a = 9;
CenComTituloSA[[a]] = "ALMACEN STC.";
CenComPreciosSA[[a]] = CenComPrec[[3, 5]];

(* PREINSTALACIONES *)
a = 10;
CenComTituloSA[[a]] = "PREINSTALACION ACA - 3";
CenComPreciosSA[[a]] = CenComPrec[[12, 5]];

a = 11;
CenComTituloSA[[a]] = "PREINSTALACION 2 B + D";
CenComPreciosSA[[a]] = CenComPrec[[11, 5]];

(* ELEMENTOS GENERALES *)
a = 12;
CenComTituloSA[[a]] = "MTU PARA CENTRALES";
CenComPreciosSA[[a]] = CenComPrec[[13, 5]];

(* ARMARIOS *)
a = 13;
CenComTituloSA[[a]] = "ARMARIO (24 BM) CON 6 ALTURAS. - LSM's";
CenComPreciosSA[[a]] = CenComPrec[[1, 5]];

a = 14;
CenComTituloSA[[a]] = "ARMARIO (24 BM) CON 6 ALTURAS. - STC's";
CenComPreciosSA[[a]] = CenComPrec[[1, 5]];
```

- Matriz de Elementos del Subsistema Selector de Grupo {CenComTituloSSG, CenComPreciosSSG, CenComDistribucionSSG, auxSSG}

```

CenComTituloSSG = Table["", {i, 1, 8}];
CenComPreciosSSG = Table["", {i, 1, 8}];
CenComDistribucionSSG = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 8}];
auxSSG = Table[0, {i, 1, 8}];

(* almacenes de conmutadores *)
a = 1;
CenComTituloSSG[[a]] = "ALM. TSM (DUP.) ENTRE 0 Y 32K";
CenComPreciosSSG[[a]] = CenComPrec[[18, 5]];

a = 2;
CenComTituloSSG[[a]] = "ALMACEN SPM (DUP.) PARA 8K";
CenComPreciosSSG[[a]] = CenComPrec[[17, 5]];

(* almacenes de procesadores regionales *)
a = 3;
CenComTituloSSG[[a]] = "ALMACEN RP (DUP.) CON 256 KW.";
CenComPreciosSSG[[a]] = CenComPrec[[14, 5]];

(* almacenes del modulo de reloj *)
a = 4;
CenComTituloSSG[[a]] = "ALM. CLM ENTRE 0 Y 32K";
CenComPreciosSSG[[a]] = CenComPrec[[15, 5]];

a = 5;
CenComTituloSSG[[a]] = "ALMACEN RCM";
CenComPreciosSSG[[a]] = CenComPrec[[16, 5]];

a = 6;
CenComTituloSSG[[a]] = "ALMACEN ICM";
CenComPreciosSSG[[a]] = CenComPrec[[20, 5]];

(* almacen del modulo de sincronismo PCM *)
a = 7;
CenComTituloSSG[[a]] = "ALMACEN PCD (DL2)";
CenComPreciosSSG[[a]] = CenComPrec[[21, 5]];

(* armarios *)
a = 8;
CenComTituloSSG[[a]] = "ARMARIO 12 BM CON 6 ALTURAS";
CenComPreciosSSG[[a]] = CenComPrec[[19, 5]];

```

■ Matriz de Elementos del Subsistema de Enlace y Señalización {CenComTituloSES, CenComPreciosSES, CenComDistribucionSES, auxSES}

```

CenComTituloSES = Table["", {i, 1, 13}];
CenComPreciosSES = Table["", {i, 1, 13}];
CenComDistribucionSES = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 13}];
auxSES = Table[0, {i, 1, 13}];

(* elementos de enlace *)
a = 1;
CenComTituloSES[[a]] = "FUNCIÓN ETC3/4 (32 CANALES)";
CenComPreciosSES[[a]] = CenComPrec[[23, 5]];

a = 2;
CenComTituloSES[[a]] = "ALMACÉN GEMM1 PARA 16 ETCC4";
CenComPreciosSES[[a]] = CenComPrec[[32, 5]];

(* elementos de señalización con centrales externas *)
a = 3;
CenComTituloSES[[a]] = "FUNCIÓN TRH (FH, PARA, ST7)";
CenComPreciosSES[[a]] = CenComPrec[[35, 5]];

a = 4;
CenComTituloSES[[a]] = "ALMACÉN TRH (FH, PARA, ST7)";
CenComPreciosSES[[a]] = CenComPrec[[34, 5]];

(* armario para elementos de enlace y señalización *)
a = 5;
CenComTituloSES[[a]] = "ARMARIO 12 BM 6 ALTURAS GEMM1 Y HLP.";
CenComPreciosSES[[a]] = CenComPrec[[33, 5]];

(* almacén para dispositivo de conferencia de llamadas *)
a = 6;
CenComTituloSES[[a]] = "ALMACÉN CCD";
CenComPreciosSES[[a]] = CenComPrec[[27, 5]];

(* armario para CCD y APZ *)
a = 7;
CenComTituloSES[[a]] = "ARMARIO (24 BM) CON 6 ALTURAS";
CenComPreciosSES[[a]] = CenComPrec[[22, 5]];

(* módulos de servidor de anuncios y expansión de memoria *)
a = 8;
CenComTituloSES[[a]] = "AST - DR V2.BÁSICO";
CenComPreciosSES[[a]] = CenComPrec[[28, 5]];

a = 9;
CenComTituloSES[[a]] = "AST - DR V2.BACK - UP";
CenComPreciosSES[[a]] = CenComPrec[[30, 5]];

a = 10;
CenComTituloSES[[a]] = "ALMACEN EMAS (4 EPROM + 4 RAM)";
CenComPreciosSES[[a]] = CenComPrec[[29, 5]];

(* módulos diversos *)
a = 11;
CenComTituloSES[[a]] = "ALMACÉN CSFSK PARA CLASS";
CenComPreciosSES[[a]] = CenComPrec[[31, 5]];

a = 12;
CenComTituloSES[[a]] = "ALM.TCON CENTRALES CON ABON.";
CenComPreciosSES[[a]] = CenComPrec[[25, 5]];

a = 13;
CenComTituloSES[[a]] = "ALMACÉN CANS (OMS)";
CenComPreciosSES[[a]] = CenComPrec[[26, 5]];

```

■ Matriz de Elementos del Subsistema de Control (APZ) {CenComTituloSC, CenComPreciosSC, CenComDistribucionSC, auxSC}

```

CenComTituloSC = Table["", {i, 1, 11}];
CenComPreciosSC = Table["", {i, 1, 11}];
CenComDistribucionSC = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 11}];
auxSC = Table[0, {i, 1, 11}];

(* dispositivos de almacenamiento *)
a = 1;
CenComTituloSC[[a]] = "FDD (FLOPPY DISK) CON 1.2 MB";
CenComPreciosSC[[a]] = CenComPrec[[37, 5]];

a = 2;
CenComTituloSC[[a]] = "HDD (HARD DISK) CON 1 GB";
CenComPreciosSC[[a]] = CenComPrec[[43, 5]];

(* elementos adicionales *)
a = 3;
CenComTituloSC[[a]] = "TARJETA RPU (INTERFAZ V - 24)";
CenComPreciosSC[[a]] = CenComPrec[[38, 5]];

a = 4;
CenComTituloSC[[a]] = "TARJETA LIU (INTERFAZ V - 24)";
CenComPreciosSC[[a]] = CenComPrec[[39, 5]];

a = 5;
CenComTituloSC[[a]] = "TARJETA ALAMP";
CenComPreciosSC[[a]] = CenComPrec[[40, 5]];

a = 6;
CenComTituloSC[[a]] = "TARJETA ALEX";
CenComPreciosSC[[a]] = CenComPrec[[41, 5]];

a = 7;
CenComTituloSC[[a]] = "TARJETA SCAN";
CenComPreciosSC[[a]] = CenComPrec[[42, 5]];

a = 8;
CenComTituloSC[[a]] = "ALMACÉN ODM";
CenComPreciosSC[[a]] = CenComPrec[[45, 5]];

(* procesador *)
a = 9;
CenComTituloSC[[a]] = "APZ 212 20";
CenComPreciosSC[[a]] = CenComPrec[[47, 5]];

a = 10;
CenComTituloSC[[a]] = "AMPLIACIÓN DE 256 MW (DUP.)";
CenComPreciosSC[[a]] = CenComPrec[[48, 5]];

(* modulo de entrada y salida *)
a = 11;
CenComTituloSC[[a]] = "IOG 11 - B5 BÁSICO";
CenComPreciosSC[[a]] = CenComPrec[[44, 5]];

```


■ Matriz de Elementos de Materiales y Equipos Diversos {CenComTituloMED, CenComPreciosMED, CenComDistribucionMED, auxMED}

```

CenComTituloMED = Table["", {i, 1, 5}];
CenComPreciosMED = Table["", {i, 1, 5}];
CenComDistribucionMED = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 5}];
auxMED = Table[0, {i, 1, 5}];

a = 1;
CenComTituloMED[[a]] = "CABLE COAXIAL (FLEX 3)";
CenComPreciosMED[[a]] = CenComPrec[[50, 5]];

a = 2;
CenComTituloMED[[a]] = "CONJ.CONECTORES PARA 1 ETCC";
CenComPreciosMED[[a]] = CenComPrec[[51, 5]];

a = 3;
CenComTituloMED[[a]] = "REGL.ABON.32 X8.CUBRET";
CenComPreciosMED[[a]] = CenComPrec[[52, 5]];

a = 4;
CenComTituloMED[[a]] = "REGL.MISCELANEOS (34 x6)";
CenComPreciosMED[[a]] = CenComPrec[[53, 5]];

a = 5;
CenComTituloMED[[a]] = "CABLE DE DISTRIB.ALIMENTACION";
CenComPreciosMED[[a]] = CenComPrec[[49, 5]];

```

■ Matriz de Elementos de Equipos de Conservación {CenComTituloEC, CenComPreciosEC, CenComDistribucionEC, auxEC}

```

CenComTituloEC = Table["", {i, 1, 3}];
CenComPreciosEC = Table["", {i, 1, 3}];
CenComDistribucionEC = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 3}];
auxEC = Table[0, {i, 1, 3}];

a = 1;
CenComTituloEC[[a]] = "MAT.CONSUMO CENTRAL APZ 212";
CenComPreciosEC[[a]] = CenComPrec[[54, 5]];

a = 2;
CenComTituloEC[[a]] = "MODULO C (PAPEL)";
CenComPreciosEC[[a]] = CenComPrec[[55, 5]];

a = 3;
CenComTituloEC[[a]] = "MODULO I (PAPEL)";
CenComPreciosEC[[a]] = CenComPrec[[56, 5]];

```

■ Matriz de Elementos de Otros Costes {CenComTituloOC, CenComPreciosOC, CenComDistribucionOC, auxOC}

```

CenComTituloOC = Table["", {i, 1, 3}];
CenComPreciosOC = Table["", {i, 1, 3}];
CenComDistribucionOC = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 3}];
auxOC = Table[0, {i, 1, 3}];

a = 1;
CenComTituloOC[[a]] = "SOFTWARE POR ACCESO BASICO";
CenComPreciosOC[[a]] = CenComPrec[[57, 5]];

a = 2;
CenComTituloOC[[a]] = "SOFTWARE POR ACCESO PRIMARIO";
CenComPreciosOC[[a]] = CenComPrec[[58, 5]];

a = 3;
CenComTituloOC[[a]] = "DDF";
CenComPreciosOC[[a]] = CenComPrec[[59, 5]];

```

■ Matriz de Elementos de una central {auxElem, auxPrec, auxTit, auxDist}

```
auxElem = Table[{auxSA, auxSSG, auxSES, auxSC, auxMED, auxEC, auxOC}];
auxPrec = Table[{CenComPreciosSA, CenComPreciosSSG,
  CenComPreciosSES, CenComPreciosSC, CenComPreciosMED, CenComPreciosEC, CenComPreciosOC}];
auxTit = Table[{CenComTituloSA, CenComTituloSSG, CenComTituloSES, CenComTituloSC,
  CenComTituloMED, CenComTituloEC, CenComTituloOC}];

auxDist = Table[{CenComDistribucionSA, CenComDistribucionSSG, CenComDistribucionSES,
  CenComDistribucionSC, CenComDistribucionMED, CenComDistribucionEC, CenComDistribucionOC}];
```

■ Matrices de Elementos de cada Subsistema de una SubCentral de una Unidad Remota

El dimensionamiento de una Central Cabecera es expresado como un patrón de diseño que agrupa los elementos necesarios agrupados de acuerdo al SubSistema al que pertenecen (Subsistema de Abonados, Subsistema Selector de Grupo, ..., Materiales y Equipos Diversos, Equipos de Conservación y Otros Costes).

Este agrupamiento no influye en los calculos, correspondiendo unicamente a una agrupacion logica de elementos de acuerdo a los subsistemas propios de una central AXE de Ericcson, así como un agrupamiento conveniente de otros elementos que intervienen en el dimensionamiento (conectores, cables, etc.).

En tal sentido dicho patrón se ha hecho extensivo tambien a las Unidades Remotas.

■ Matriz de Elementos del Subsistema de Abonado {CenComTituloSAr, CenComPreciosSAr, CenComDistribucionSAr, auxSAr}

```

CenComTituloSAr = Table["", {i, 1, 20}];
CenComPreciosSAr = Table[0, {i, 1, 20}];
CenComDistribucionSAr = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 20}];
auxSAr = Table[0, {i, 1, 20}];

(* TARJETAS DE LINEA *)
a = 1;
CenComTituloSAr[[a]] = "TARJETA PARA 8 ABONADOS ACA-3.";
CenComPreciosSAr[[a]] = CenComPrec[[76, 5]];

a = 2;
CenComTituloSAr[[a]] = "TARJETA PARA 4 ACCESOS 2B+D.";
CenComPreciosSAr[[a]] = CenComPrec[[63, 5]];

a = 3;
CenComTituloSAr[[a]] = "TARJETA KR PARA LSM DE ACA - 3";
CenComPreciosSAr[[a]] = CenComPrec[[89, 5]];

(* LSM's *)
a = 4;
CenComTituloSAr[[a]] = "LSM ACA-3 128. STR,KR,ETB,POU.";
CenComPreciosSAr[[a]] = CenComPrec[[71, 5]];

a = 5;
CenComTituloSAr[[a]] = "LSM ACA-3 128. KR,ETB,POU.";
CenComPreciosSAr[[a]] = CenComPrec[[72, 5]];

a = 6;
CenComTituloSAr[[a]] = "LSM ACA-3 128. ETB,POU.";
CenComPreciosSAr[[a]] = CenComPrec[[73, 5]];

a = 7;
CenComTituloSAr[[a]] = "LSM ACA-3 128. POU.";
CenComPreciosSAr[[a]] = CenComPrec[[74, 5]];

a = 8;
CenComTituloSAr[[a]] = "LSM ACA-3 PARA 128 AB. POU.";
CenComPreciosSAr[[a]] = CenComPrec[[75, 5]];

a = 9;
CenComTituloSAr[[a]] = "LSM PARA 64 ACC. 2B+D. INCLUYE 8 ACCESOS";
CenComPreciosSAr[[a]] = CenComPrec[[66, 5]];

a = 10;
CenComTituloSAr[[a]] = "LSM PARA 64 ACC. 2B+D. INCLUYE 16 ACCESOS";
CenComPreciosSAr[[a]] = CenComPrec[[67, 5]];

a = 11;
CenComTituloSAr[[a]] = "LSM PARA 64 ACC. 2B+D. INCLUYE 32 ACCESOS";
CenComPreciosSAr[[a]] = CenComPrec[[68, 5]];

a = 12;
CenComTituloSAr[[a]] = "LSM PARA 64 ACC. 2B+D. INCLUYE 48 ACCESOS";
CenComPreciosSAr[[a]] = CenComPrec[[69, 5]];

a = 13;
CenComTituloSAr[[a]] = "LSM PARA 64 ACC. 2B+D. INCLUYE 60 ACCESOS";
CenComPreciosSAr[[a]] = CenComPrec[[70, 5]];

(* ELEMENTOS DE CONEXION CON GSS *)
a = 14;
CenComTituloSAr[[a]] = "TARJETA ETB PARA R.D.S.I.";
CenComPreciosSAr[[a]] = CenComPrec[[65, 5]];

a = 15;
CenComTituloSAr[[a]] = "TARJETA STR PARA LSM DE ACA-3.";
CenComPreciosSAr[[a]] = CenComPrec[[77, 5]];

a = 16;
CenComTituloSAr[[a]] = "TARJETA STR PARA R.D.S.I.";
CenComPreciosSAr[[a]] = CenComPrec[[64, 5]];

(* PREINSTALACIONES *)
a = 17;
CenComTituloSAr[[a]] = "PREINSTALACION ACA-3";

```

```
CenComPreciosSAr[[a]] = CenComPrec[[77, 5]];

a = 18;
CenComTituloSAr[[a]] = "PREINSTALACION 2B+D";
CenComPreciosSAr[[a]] = CenComPrec[[80, 5]];

(* ELEMENTOS GENERALES *)
a = 19;
CenComTituloSAr[[a]] = "ALMACEN MTU PARA RSS";
CenComPreciosSAr[[a]] = CenComPrec[[78, 5]];

(* ARMARIOS *)
a = 20;
CenComTituloSAr[[a]] = "ARMARIO (24 BM) CON 6 ALTURAS. - RSS";
CenComPreciosSAr[[a]] = CenComPrec[[62, 5]];
```

■ Matriz de Elementos de Materiales y Equipos Diversos {CenComTituloMEDr, CenComPreciosMEDr, CenComDistribucionMEDr, auxMEDr}

```
CenComTituloMEDr = Table["", {i, 1, 4}];
CenComPreciosMEDr = Table["", {i, 1, 4}];
CenComDistribucionMEDr = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 4}];
auxMEDr = Table[0, {i, 1, 4}];

a = 1;
CenComTituloMEDr[[a]] = "CONECTOR PARA FLEX 3.";
CenComPreciosMEDr[[a]] = CenComPrec[[82, 5]];

a = 2;
CenComTituloMEDr[[a]] = "REGL. MISCELANEOS (34x6).";
CenComPreciosMEDr[[a]] = CenComPrec[[84, 5]];

a = 3;
CenComTituloMEDr[[a]] = "PUENTE CON PUNTO DE PRUEBA.";
CenComPreciosMEDr[[a]] = CenComPrec[[83, 5]];

a = 4;
CenComTituloMEDr[[a]] = "CABLE DE DISTRIB.ALIMENTACION";
CenComPreciosMEDr[[a]] = CenComPrec[[81, 5]];
```

■ Matriz de Elementos de Equipos de Conservación {CenComTituloECr, CenComPreciosECr, CenComDistribucionECr, auxECr}

```
CenComTituloECr = Table["", {i, 1, 3}];
CenComPreciosECr = Table["", {i, 1, 3}];
CenComDistribucionECr = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 3}];
auxECr = Table[0, {i, 1, 3}];

a = 1;
CenComTituloECr[[a]] = "MODULO C (PAPEL) PARA RSS.";
CenComPreciosECr[[a]] = CenComPrec[[85, 5]];

a = 2;
CenComTituloECr[[a]] = "MODULO I (PAPEL) PARA RSS.";
CenComPreciosECr[[a]] = CenComPrec[[86, 5]];

a = 3;
CenComTituloECr[[a]] = "DOC. PARA RSS NUEVOS.";
CenComPreciosECr[[a]] = CenComPrec[[87, 5]];
```

■ Matriz de Elementos de Otros Costes {CenComTituloOCr, CenComPreciosOCr, CenComDistribucionOCr, auxOCr}

```
CenComTituloOCr = Table["", {i, 1, 2}];
CenComPreciosOCr = Table["", {i, 1, 2}];
CenComDistribucionOCr = Table[Table[0, {j, 1, Length[CenComCatCosto]}], {i, 1, 2}];
auxOCr = Table[0, {i, 1, 2}];

a = 1;
CenComTituloOCr[[a]] = "SOFTWARE POR ACCESO BASICO";
CenComPreciosOCr[[a]] = CenComPrec[[88, 5]];

a = 2;
CenComTituloOCr[[a]] = "DDF";
CenComPreciosOCr[[a]] = CenComPrec[[90, 5]];
```

■ Matriz de Elementos de una Unidad Remota {auxElemr, auxPrecr, auxTitr, auxDisrt}

```
auxElemr = Table[{auxSAr, auxMEDr, auxECr, auxOCr}];
auxPrecr = Table[{CenComPreciosSAr, CenComPreciosMEDr, CenComPreciosECr, CenComPreciosOCr}];
auxTitr = Table[{CenComTituloSAr, CenComTituloMEDr, CenComTituloECr, CenComTituloOCr}];

auxDistr = Table[{CenComDistribucionSAr, CenComDistribucionMEDr, CenComDistribucionECr, CenComDistribucionOCr}];
```

■ Matriz de Elementos de todas las Centrales Cabecera y Unidades Remotas {CenComElemCab, CenComElemRem, CenComAbonSubCentRem}

```
(* Elementos para cada Cabecera*)
CenComElemCab = Table[auxElem, {i, 1, CenComN}];

(* Elementos de cada SubCentral para cada Unidad Remota*)
CenComElemRem = Table[{auxElemr}, {i, 1, CenComN}];
(* Numero de Abonados por Tipo (PSTN y RDSI Basico )
y Enlaces (Eos') de cada SubCentral para cada Unidad Remota *)
auxAbonSubCent = {0, 0, 0};
CenComAbonSubCentRem = Table[{auxAbonSubCent}, {i, 1, CenComN}];
```

■ Tablas Auxiliares de dimensionamiento de armarios

Los almacenes ("magazines" o "shelves") se alojan en armarios. dichos armarios tienen diferentes capacidades de acuerdo a cada tipo de almacen (LSM, TSM, etc.). Se incluye el dimensionamiento específico del número de armarios requeridos de acuerdo al número de LSM's, TSM's y SPM's, de las centrales Cabecera. Dicho dimensionamiento está expresado como una matriz que relaciona el número de almacenes y el número de armarios necesarios. Estas matrices son utilizadas normalmente mediante la función auxiliar "BuscarV"

■ Dimensionamiento de Armarios para LSM's {CenComMLSMArmarios}

```
CenComMLSMArmarios = {
  {0 , 0} ,
  {16 , 2},
  {32 , 3},
  {48 , 5},
  {64 , 6},
  {80 , 8},
  {96 , 9},
  {112, 11},
  {128, 12},
  {144 , 14},
  {160 , 15},
  {176 , 17},
  {192, 18},
  {208 , 20},
  {224 , 21},
  {240 , 23},
  {256 , 24}
};
```


■ Dimensionamiento de Armarios para TSM {CenComMTSMArmarios}

```
CenComMTSMArmarios = {  
  {0, 0},  
  {8, 2},  
  {16, 3},  
  {24, 5},  
  {32, 6},  
  {40, 8},  
  {48, 9},  
  {56, 11},  
  {64, 12},  
  {72, 14},  
  {80, 15},  
  {88, 17},  
  {96, 18},  
  {104, 20},  
  {112, 21},  
  {120, 23},  
  {128, 24}  
};
```

■ Dimensionamiento de armarios para SPM {CenComMSPMArmarios}

```
CenComMSPMArmarios = {  
  {0, 0},  
  {2, 1},  
  {8, 4},  
  {16, 9},  
  {32, 16}  
};
```

■ Funciones de Dimensionamiento de Centrales

■ Dimensionar Cabeceras {DimensionaCabecera}

```

DimensionaCabecera[mNumPSTN_, mNumRDSIBA_, mNumRDSIPRI_, mNumRem_, mNumE0_] :=
Module[{auxElem, auxSA, auxSSG, auxSES, auxSC, auxMED, auxEC, auxOC, i, j, k,
  auxEspacioLSMs, AuxArmariosLSMs, AuxArmariosSTC, AuxArmariosTSM, AuxArmariosSPM},
  auxSA = Table[0, {i, 1, 14}];
  auxSSG = Table[0, {i, 1, 8}];
  auxSES = Table[0, {i, 1, 13}];
  auxSC = Table[0, {i, 1, 11}];
  auxMED = Table[0, {i, 1, 5}];
  auxEC = Table[0, {i, 1, 3}];
  auxOC = Table[0, {i, 1, 3}];

  (* ----- *)
  (*Sistema de Abonado*)

  (*Tarjeta de Abonado*)
  auxSA[[1]] = 0;
  (*Tarjetas KR*)
  auxSA[[2]] = RedondearMas[ $\frac{2 * mNumPSTN}{2048}$ ];

  (*LSM PSTN*)
  auxSA[[3]] = RedondearMas[ $\frac{mNumPSTN}{128}$ ];
  (*LSM RDSI BA*)
  auxSA[[4]] = RedondearMas[ $\frac{mNumRDSIBA}{60}$ ];
  (*LSM RDSI PRI*)
  auxSA[[5]] = RedondearMas[ $\frac{mNumRDSIPRI}{4}$ ];

  (*almacen RP - Procesadores Reginales RPBC Duplicado *)
  auxSA[[6]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0, 0, RedondearMas[ $\frac{auxSA[[3]] + auxSA[[4]] + auxSA[[5]]}{16}$ ]];
  (*almacen SE-PRM - Equipo Especial de Medicion de Pulsos Remotos*)
  auxSA[[7]] = 0; (* =If[AñoDeEjecucion==2000,RedondearMas[0.00008*mNumPSTN],0] *)

  (*tarjetas JTC*)
  auxSA[[8]] = RedondearMas[auxSA[[4]] +  $\frac{auxSA[[3]]}{2}$ ];
  (*almacen STC - Señalizadores para Unidades Remotas*)
  auxSA[[9]] = RedondearMas[mNumRem * 1.6934];

  (*preinstalacion PSTN Preinstalacion ACA-3*)
  auxSA[[10]] = RedondearMas[auxSA[[3]] * 16 - auxSA[[1]]];
  (*preinstalacion basicos 2b+d*)
  auxSA[[11]] = auxSA[[4]] * 16 - RedondearMas[ $\frac{mNumRDSIBA}{4}$ ];
  (*MTU para centrales*)
  auxSA[[12]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0, 0, 1];

  (*Armario 24 BM 6 alturas*)
  auxEspacioLSMs = Ceiling[auxSA[[3]] + auxSA[[4]] * 2 + auxSA[[5]], 16];
  AuxArmariosLSMs = BuscarV[auxEspacioLSMs, CenComMLSMArmarios, 2];
  AuxArmariosSTC = RedondearMas[ $\frac{auxSA[[9]]}{32}$ ];
  auxSA[[13]] = AuxArmariosLSMs;
  auxSA[[14]] = AuxArmariosSTC;

  (* ----- *)
  (*Sistema de Control - APZ*)

  (* Disketera *)
  auxSC[[1]] = 2;
  (* Disco duro *)
  auxSC[[2]] = 1;
  (* Tarjeta RPU *)
  auxSC[[3]] = 4;
  (* Tarjeta LIU *)
  auxSC[[4]] = 7;
  (* Tarjeta ALAMP *)
  auxSC[[5]] = 1;
  (* Tarjeta ALEX *)

```

```

auxSC[[6]] = 1;
(* Tarjeta SCAN *)
auxSC[[7]] = 1;
(* Almacen ODM *)
auxSC[[8]] = 1;
(* APZ 212 20 *)
auxSC[[9]] = 1;
(* Ampliacion memoria 256 MW Dup *)
auxSC[[10]] = 0;
(* Conjunto IOG 11 - B5 Basico *)
auxSC[[11]] = 1;

(* ----- *)
(*Subsistema de Enlace y Señalización *)

(* Tarjetas ETC - Tarjeta de Enlace de 1 E1 = 30 E0's Funcion ETC *)
auxSES[[1]] = RedondearMas $\left[\frac{mNumE0}{30}\right]$ ;
(* Almacén para Tarjetas ETC - Almacen GEMM1 para 16 Tarjetas ETC *)
auxSES[[2]] = RedondearMas $\left[\frac{auxSES[[1]]}{16}\right]$ ;

(* Tarjeta TRH - Funcion TRH (FH, PRA, ST7) - Transaction Record Handler,
Frame Handler, RSDI PRI, Señalización SS7 *)
auxSES[[3]] = RedondearMas $\left[\frac{mNumRDSIPRI}{16}\right]$  + RedondearMas $\left[\frac{auxSA[[4]]}{12}\right]$  + RedondearMas[mNumE0 * 0.0007];

(* Almacen para Tarjeta TRH*)
auxSES[[4]] = RedondearMas $\left[\frac{auxSES[[3]]}{4}\right]$ ;

(*Armario 12 BM 6 alturas para tarjetas ETC - GEMM1 y HLP*)
auxSES[[5]] = RedondearMas $\left[\frac{auxSES[[2]] + auxSES[[4]]}{6}\right]$ ;

(* Almacen CCD - Call Conference Device *)
auxSES[[6]] = RedondearMas $\left[\frac{mNumPSTN * 0.1}{128}\right]$ ;

(* Armario 24 BM 6 alturas *)
auxSES[[7]] = RedondearMas $\left[\frac{auxSES[[6]]}{8}\right]$  + 2 * auxSC[[9]];

(* AST - DR V2 Basico *)
auxSES[[8]] = 1;
(* AST - DR V2 Backup *)
auxSES[[9]] = 1;
(* Almacen EMAS *)
auxSES[[10]] = auxSES[[9]];

(* Almacen CSFSK para CLASS*)
auxSES[[11]] = 1;
(* Almacen TCON *)
auxSES[[12]] = 1;
(* Almacen CANS - incluye OMS *)
auxSES[[13]] = 1;

(* ----- *)
(*Subsistema Selector de Grupo *)

(* Almacen CLM - entre 0 y 32 k*)
auxSSG[[4]] = 3;
(* Almacen RCM *)
auxSSG[[5]] = 1;
(* Almacen ICM *)
auxSSG[[6]] = 2;
(* Almacen PCD - DL2 *)
auxSSG[[7]] = 2;

(* Almacen de Conmutadores Temporales - TSM Dup entre 0 y 32 k*)
auxSSG[[1]] = RedondearMas $\left[\frac{1}{16} (auxSES[[2]] * 16 + auxSES[[6]] + \right.$ 
 $\left. auxSA[[8]] + auxSES[[8]] * 2 + auxSSG[[7]] + auxSES[[3]] + auxSES[[11]] + auxSES[[12]])\right]$ ;
(* Almacen de Conmutadores Espaciales - SPM Dup para 8k *)
auxSSG[[2]] =
If $\left[\text{Ceiling}[auxSSG[[1]] * 0.5, 16] == 16, \frac{4}{2},$ 

```

```

    If[Ceiling[auxSSG[[1]] * 0.5, 16] == 32,  $\frac{16}{2}$ ,
    If[Ceiling[auxSSG[[1]] * 0.5, 16] == 48,  $\frac{36}{2}$ ,
    If[Ceiling[auxSSG[[1]] * 0.5, 16] == 64,  $\frac{62}{2}$ , 32]
    ]
  ]
];

(* almacen de Procesadores Regionales - RP Dup con 256 KW*)
auxSSG[[3]] = RedondearMas[ $\frac{\text{auxSSG}[[1]]}{8}$ ] + 1 + RedondearMas[ $\frac{\text{auxSES}[[6]]}{8}$ ] + auxSES[[12]] + 1;

(*Armario 12 BM 6 alturas*)
AuxArmariosTSM = BuscarV[Ceiling[auxSSG[[1]], 8], CenComMTSMArmarios, 2];
AuxArmariosSPM = BuscarV[auxSSG[[2]], CenComMSPMArmarios, 2];
auxSSG[[8]] = AuxArmariosTSM + AuxArmariosSPM + 1;

(* ----- *)
(* Materiales y Equipos Diversos *)

(* cable coaxial flex3 *)
auxMED[[1]] = RedondearMas[1.5898 * mNumE0];
(* conectores para ETC *)
auxMED[[2]] = auxSES[[1]];
(* regleta de abonados 32 x 8 cubret *)
auxMED[[3]] = RedondearMas[ $\frac{\text{mNumPSTN}}{128}$ ] + RedondearMas[ $\frac{\text{mNumRDSIBA}}{128}$ ];
(* regleta miscelaneos 34 x 6 *)
auxMED[[4]] = 2;
(* cable de distribucion de alimentacion *)
auxMED[[5]] = RedondearMas[0.6524 * (mNumPSTN + 2 * mNumRDSIBA + 30 * mNumRDSIPRI + mNumE0)];

(* ----- *)
(* Equipos de Conservacion *)

(* Materiales de Consumo Central APZ 212 *)
auxEC[[1]] = 1;
(* Modulo C *)
auxEC[[2]] = 1;
(* Modulo I *)
auxEC[[3]] = 1;

(* ----- *)
(* Otros Costes *)

(* Software por Acceso Basico *)
auxOC[[1]] = Ceiling[mNumRDSIBA, 64];
(* Software por Acceso Primario *)
auxOC[[2]] = mNumRDSIPRI;
(* DDF *)
auxOC[[3]] = 1;

(* Crea Matriz Con Resultado Final del Dimensionamiento*)
auxElem = Table[{auxSA, auxSSG, auxSES, auxSC, auxMED, auxEC, auxOC}];
Return[auxElem];

];
```

■ Dimensionar Unidades Remotas {DimensionaSubCentral, CenComNumAbonadoSSS, DimensionaAbonadosPorSubCentral, DimensionaRemota}

```

DimensionaSubCentral[mNumPSTN_, mNumRDSIBA_, mNumE0_, mNumSubCentral_] := Module[{auxElemr, auxSAr,
  auxMEDr, auxECr, auxOCr, i, j, k, auxNumeroLSMs, AuxArmariosLSMs, auxPreInstalacionRSDIBasicos, a},
auxSAr = Table[0, {i, 1, 20}];
auxMEDr = Table[0, {i, 1, 4}];
auxECr = Table[0, {i, 1, 3}];
auxOCr = Table[0, {i, 1, 2}];

(* ----- *)
(*Sistema de Abonado*)

(*LSM PSTN*)
auxSAr[[4]] = 0;
```

```

auxSAr[[5]] = 0;
auxSAr[[6]] = 0;

auxSAr[[7]] = RedondearMas $\left[\frac{\text{mNumPSTN}}{128}\right]$ ;

auxSAr[[8]] = 0;

(*LSM RDSI BA*)
If[(mNumRDSIBA > 0) && (mNumRDSIBA < 16), a = 1;, a = 0];
auxSAr[[9]] = a;

If[(mNumRDSIBA ≥ 16) && (mNumRDSIBA < 32), a = 1, a = 0];
auxSAr[[10]] = a;

If[(mNumRDSIBA ≥ 32) && (mNumRDSIBA < 48), a = 1, a = 0];
auxSAr[[11]] = a;

If[(mNumRDSIBA ≥ 48) && (mNumRDSIBA < 60), a = 1, a = 0];
auxSAr[[12]] = a;

If $\left[\text{mNumRDSIBA} \geq 60, a = \text{RedondearMas}\left[\frac{\text{mNumRDSIBA}}{64}\right], a = 0\right]$ ;
auxSAr[[13]] = a;

(*Armario 24 BM 6 alturas*)
auxNumeroLSMs = 2 * (auxSAr[[9]] + auxSAr[[10]] + auxSAr[[11]] + auxSAr[[12]] + auxSAr[[13]]) +
  auxSAr[[4]] + auxSAr[[5]] + auxSAr[[6]] + auxSAr[[7]] + auxSAr[[8]];
If[(mNumPSTN + mNumRDSIBA) > 0,
  If[(auxNumeroLSMs > 16), AuxArmariosLSMs = 3,
    If[(auxNumeroLSMs > 6), AuxArmariosLSMs = 2, AuxArmariosLSMs = 1]];
  , AuxArmariosLSMs = 0];
auxSAr[[20]] = AuxArmariosLSMs;

(*Tarjeta de Abonado RSDI Basico*)
If $\left[(\text{mNumRDSIBA} < 8),\right.$ 
  a = RedondearMenos $\left[\frac{1}{4} (\text{mNumRDSIBA} - (\text{auxSAr}[[9]] * 8 + \text{auxSAr}[[10]] * 16 + \text{auxSAr}[[11]] * 32 + \text{auxSAr}[[12]] * 48 + \text{auxSAr}[[13]] * 60))\right]$ ;
  ,
  a = RedondearMas $\left[\frac{1}{4} (\text{mNumRDSIBA} - (\text{auxSAr}[[9]] * 8 + \text{auxSAr}[[10]] * 16 + \text{auxSAr}[[11]] * 32 + \text{auxSAr}[[12]] * 48 + \text{auxSAr}[[13]] * 60))\right]$ ;
 $\left. \right]$ ;
auxSAr[[2]] = a;

(* tarjetas de conexion con GSS *)
(* tarjeta str para rdsi *)
If[(mNumPSTN + 2 * mNumRDSIBA + mNumE0) > 0,
  If[auxSAr[[4]] == 1,
    a = 1,
    If[auxSAr[[4]] == 0, a = 2, a = 0]];
  ,
  a = 0];
auxSAr[[16]] = a;

(* tarjeta etb para rdsi *)
auxSAr[[14]] = RedondearMas $\left[\frac{\text{mNumE0}}{30}\right]$ ; (* -auxSAr[[4]]-auxSAr[[6]] *)

(*Tarjeta de Abonado PSTN*)
auxSAr[[1]] = 0;

(* tarjeta str para pstn *)
auxSAr[[15]] = 0;

(*Almacen MTU para RSS*)
(*If[mNumSubCentral==1,
  If[( mNumPSTN+mNumRDSIBA+mNumE0 )>0,a=1,a=0]
  ,a=0]; *)
If[( mNumPSTN + mNumRDSIBA + mNumE0 ) > 0, a = 1, a = 0];
auxSAr[[19]] = a;

(*preinstalacion PSTN*)
auxSAr[[17]] = -1 * (auxSAr[[1]]);

(*preinstalacion basicos 2b+d*)
auxPreInstalacionRSDIBasicos =

```



```
RedondearMas $\left[\frac{1}{4} (60 * (auxSAr[[9]] + auxSAr[[10]] + auxSAr[[11]] + auxSAr[[12]] + auxSAr[[13]]) - mNumRDSIBA)\right];$ 

auxSAr[[18]] = auxPreInstalacionRSDIBasicos;

(*Tarjetas KR*)
If[mNumPSTN == 0,
  a = 0,
  If[(auxSAr[[4]] + auxSAr[[5]]) == 0,
    a = 1;,
    a = 0;]
];
auxSAr[[3]] = a;

(* ----- *)
(* Materiales y Equipos Diversos *)
(* cable de distribucion de alimentacion *)

If[(mNumPSTN + 2 * mNumRDSIBA + mNumE0) > 0
,
  If[AuxArmariosLSMs == 1,
    a = 160,
    If[AuxArmariosLSMs == 2,
      a = 360, If[AuxArmariosLSMs == 3, a = 480, a = 600]; ]
,
  a = 0
];
];
auxMEDr[[4]] = a;

(* conectores para flex *)
auxMEDr[[1]] = 2 * (auxSAr[[14]] + auxSAr[[4]] + auxSAr[[5]] + auxSAr[[6]] );

(* puente con punto de prueba *)
auxMEDr[[3]] = auxMEDr[[1]];

(* regleta miscelaneos 34 x 6 *)
If[(mNumPSTN + mNumRDSIBA + mNumE0) > 0, a = 1, a = 0];
auxMEDr[[2]] = a;

(* ----- *)
(* Equipos de Conservacion *)

(* MODULO C (PAPEL) PARA RSS. *)
auxECr[[1]] = 0;

(* MODULO I (PAPEL) PARA RSS. *)
If[mNumSubCentral == 1,
  If[(mNumPSTN + mNumRDSIBA + mNumE0) > 0, a = 1, a = 0]
,
  a = 0];
auxECr[[2]] = a;

(* DOC.PARA RSS NUEVOS. *)
If[mNumSubCentral == 1,
  If[(mNumPSTN + mNumRDSIBA + mNumE0) > 0, a = 1, a = 0];
,
  a = 0];
auxECr[[3]] = a;

(* ----- *)
(* Otros Costes *)

(* Software por Acceso Basico *)
If[mNumRDSIBA == 0,
  a = 0
,
  a = 64 * RedondearMas $\left[\frac{mNumRDSIBA}{64}\right];$ 
];
auxOCr[[1]] = a;

(* DDF *)
If[mNumSubCentral == 1,
  If[(mNumPSTN + mNumRDSIBA + mNumE0) > 0, a = 1, a = 0];
,
  a = 0];
auxOCr[[2]] = a;
```

```
(* Crea Matriz Con Resultado Final del Dimensionamiento*)
auxElemr = Table[{auxSAr, auxMEDr, auxECr, auxOCr}];
Return[auxElemr];

];
```

```
(* ----- *)
(* FUNCION AUXILIAR QUE SUMA EL NUMERO DE ABONADOS DE UN SSS DE ACUERDO AL TIPO DE LSM (PSTN o BA)
PRIMERO SELECCIONA LOS LSM'S QUE SON DEL TIPO INDICADO
LUEGO SUMA EL SEGUNDO ELEMENTO (NUMERO DE ABONADOS) Y LO DEVUELVE COMO RESULTADO
*)
CenComNumAbonadoSSS[SSS_, TipoLSM_] :=
Module[
{auxLSMsTipo, AuxSuma, i, x, y},
auxLSMsTipo = Cases[SSS, {x1_, y1_} /; x1 == TipoLSM];
AuxSuma =  $\sum_{i=1}^{\text{Length}[auxLSMsTipo]}$  auxLSMsTipo[[i, 2]];
Return[AuxSuma];
];
```

```
(* Esta funcion devuelve un vector con el numero
de abonados (PSTN-Basicos) y Enlaces (E0s) para cada subcentral
En el caso que no existan abonados ni enlaces devuelve una matriz conteniendo una unica central vacia *)

DimensionaAbonadosPorSubCentral[ mNumPSTN_, mNumRDSIBA_, mNumE0_] :=
Module[
{ mauxElemr,
auxLSMxSSS,
auxAbonPSTNxLSMPSTN, auxAbonBAXLSMBA,
auxNumLSMPSTN, auxNumLSMBA,
mListaLSM,
mListaLSMBA, mListaLSMPSTN,
mNumSSS, mNumLSMs, auxLSMVacio,
mAuxAbonE0xSSS,
auxNumE0xSSS,
auxA}
,
mauxElemr = {auxAbonSubCent};

(* validacion para unidad remota sin elementos *)
If[(mNumPSTN + mNumRDSIBA + mNumE0) == 0, Return[mauxElemr]];

(*-----*)
(* auxiliares *)
auxLSMVacio = {"PSTN", 0};

(* equivalencias *)
(* una subcentral equivale a un SSS o modulo de conmutacion de abonados *)

(* lsm's por sss *)
auxLSMxSSS = 16;
(* abonados por tipo de lsm *)
auxAbonPSTNxLSMPSTN = 128;
auxAbonBAXLSMBA = 64;

(* priorizacion de llenado de SSS
1 BA;
2 PSTN;
*)

(*-----*)
(* calcula el numero de abonados por SSS *)

(* numero de lsm's por servicio *)
auxNumLSMPSTN = Ceiling[ $\frac{mNumPSTN}{auxAbonPSTNxLSMPSTN}$ ];
auxNumLSMBA = Ceiling[ $\frac{mNumRDSIBA}{auxAbonBAXLSMBA}$ ];
```

```

(* crea lista ordenada de LSM's en funcion a su tipo
   y adiciona el numero de abonados *)
mListaLSMBA = Table[{"BA", auxAbonBAxLSMBA}, {auxNumLSMBA}] ;
If[Mod[mNumRDSIBA, auxAbonBAxLSMBA] > 0, mListaLSMBA[[auxNumLSMBA, 2]] = Mod[mNumRDSIBA, auxAbonBAxLSMBA];
];

mListaLSMPSTN = Table[{"PSTN", auxAbonPSTNxLSMPSTN}, {auxNumLSMPSTN}] ;
If[Mod[mNumPSTN, auxAbonPSTNxLSMPSTN] > 0,
  mListaLSMPSTN[[auxNumLSMPSTN, 2]] = Mod[mNumPSTN, auxAbonPSTNxLSMPSTN];
];

mListaLSM = Table[{ mListaLSMBA, mListaLSMPSTN}];
mListaLSM = Flatten[mListaLSM, 1];
mNumLSMs = Length[mListaLSM];

(* calcula numero de subcentrales *)
mNumSSS = Ceiling[ $\frac{mNumLSMs}{auxLSMxSSS}$ ];

(* completa bloques de subcentrales pra subdividir las posteriormente *)
mListaLSM = PadRight[mListaLSM, mNumSSS * auxLSMxSSS, {auxLSMVacio}];
mListaLSM = Partition[mListaLSM, auxLSMxSSS];

(* variable para almacenar division de subcentrales *)
(*
mAuxAbonE0xSSS;
1 = Num PSTN;
2 = Num RSDI Basico;
3 = Num Enlaces E0's;
*)
mAuxAbonE0xSSS = Table[{CenComNumAbonadoSSS[mListaLSM[[i]], "PSTN"], CenComNumAbonadoSSS[mListaLSM[[i]], "BA"],
  0}, {i, 1, Length[mListaLSM]}];

(*-----*)
(* calcula el numero de enlaces por SSS *)
(* el numero de enlaces por sss se calcula prorrateando el numero de enlaces en funcion al numero
   de abonados equivalentes de cada sss *)
Do[mAuxAbonE0xSSS[[i, 3]] = RedondearMenos[ $\frac{mNumE0 * (mAuxAbonE0xSSS[[i, 1]] + mAuxAbonE0xSSS[[i, 2]] * 2)}{(mNumPSTN + mNumRDSIBA * 2)}$ ],
  {i, 1, Length[mAuxAbonE0xSSS]}];
(* corrige el ultimo elemento con el numero de enlaces faltantes *)
auxA =  $\sum_{i=1}^{Length[mAuxAbonE0xSSS]-1} (mAuxAbonE0xSSS[[i, 3]]);$ 
mAuxAbonE0xSSS[[Length[mAuxAbonE0xSSS], 3]] = mNumE0 - auxA;

Return[mAuxAbonE0xSSS];
];

```

```
(* ----- *)
(* Esta funcion recibe como entrada un vector que contiene el numero
de abonados(PSTN-BA) y Enlaces (E0s) para cada Subcentral de una unidad Remota.
Devuelve una matriz de elementos dimensionados para cada subcentral.
En el caso que no existan abonados ni enlaces devuelve una matriz conteniendo una unica central vacia. *)
DimensionaRemota[mAbonSubCent_] :=
Module[
{iauxElemr,
mNumPSTN, mNumRDSIBA, mNumE0,
mAuxAbonE0xSSS,
auxA
},

iauxElemr = {iauxElemr};

(* validacion para unidad remota sin SubCentrales *)
If[Length[mAbonSubCent] == 0, Return[iauxElemr]];

(*-----*)
(* dimensiona cada subcentral de acuerdo al numero de abonados y e0's*)

iauxElemr = Table[
DimensionaSubCentral[
mAbonSubCent[[i, 1]],
mAbonSubCent[[i, 2]],
mAbonSubCent[[i, 3]],
i
]

, {i, 1, Length[mAbonSubCent]}
];

Return[iauxElemr];

];
```

■ Cantidad de Materiales {CenComElemCab, CenComElemRem}

Se calcula el dimensionamiento (numero de elementos) necesarios para cada Cabecera y Unidad Remota.

En el caso de las Unidades Remotas se calcula el dimensionamiento por cada SubCentral

```
Do[
NumPSTN = CenComInput[[i, 6]];
NumRDSIBA = CenComInput[[i, 7]];
NumRDSIPRI = CenComInput[[i, 8]];
NumRem = CenComRemCab[[i, 2]];
NumE0 = CenComE0s[[i]];
If[ CenComInput[[i, 5]] == "CABECERA",
(* DIMENSIONA CABECERAS*)
CenComElemCab[[i]] = DimensionaCabecera[NumPSTN, NumRDSIBA, NumRDSIPRI, NumRem, NumE0];
,
(*DIMENSIONA UNIDADES REMOTAS*)
(* calcula el numero de abonados y enlaces por cada subcentral *)
CenComAbonSubCentRem[[i]] = DimensionaAbonadosPorSubCentral[NumPSTN, NumRDSIBA, NumE0];
(* calcula el numero de elementos por cada subcentral *)
CenComElemRem[[i]] = DimensionaRemota[CenComAbonSubCentRem[[i]] ];

];
,
{i, 1, CenComN}];
```

■ Costos de Materiales {CenComMatCab,CenComMatRem}

```
(* Costos de Materiales para Cabeceras *)
CenComMatCab = Table[auxElem, {i, 1, CenComN}];

Do[
  CenComMatCab[[i]] = CenComElemCab[[i]] * auxPrec;
  ,
  {i, 1, CenComN}
];

(* Costos de Materiales para Unidades Remotas *)
CenComMatRem = Table[{auxElemr}, {i, 1, CenComN}];

Do[
  (* unidad remota *)
  AuxR = CenComElemRem[[i]];
  (* calcula costo para cada subcentral *)
  AuxCostR = Table[(AuxR[[j]] * auxPrecr), {j, 1, Length[AuxR]}];
  (* asigna costo calculado *)
  CenComMatRem[[i]] = AuxCostR;
  ,
  {i, 1, CenComN}
];
```

■ Costos de Instalación, Descuentos de Costo de instalación para Unidades Remotas y No Aplicación de Costo de instalacion {CenComInstCab,CenComInstRem}

```
(* costo de instalacion para Cabeceras *)
CenComInstCab = CenComMatCab * CenComPorcInst;

(* elimina aplicacion de factor de instalacion a elementos identificados *)
Do[
  (* Equipos de Conservación *)
  CenComInstCab[[i, 6]] = CenComInstCab[[i, 6]] * 0;
  ,
  {i, 1, CenComN}
];

(* costo de instalacion y descuentos para Unidades Remotas *)
CenComInstRem = Table[{auxElemr}, {i, 1, CenComN}];
(* En las Unidades Remotas se considera un descuento por instalacion
este descuento depende del numero de abonados (PSTN)
se debe calcular este descuento para cada Subcentral de cada Unidad Remota *)

Do[
  (* numero de abonados y enlaces de unidad remota *)
  AuxAbonSCR = CenComAbonSubCentRem[[i]];
  AuxNumSubCent = Length[AuxAbonSCR];

  (* costo de materiales de unidad remota *)
  AuxCostMatR = CenComMatRem[[i]];
  AuxCostIntR = Table[{auxElemr}, {j, 1, AuxNumSubCent}];

  (* calcula descuento para cada subcentral *)
  Do[

    AuxNumPSTN = AuxAbonSCR[[j, 1]];
    If[AuxNumPSTN ≤ 768, AuxDcto = 0.928, AuxDcto = 0.876];

    AuxCostMatSubCent = AuxCostMatR[[j]];

    AuxCostIntR[[j]] = AuxCostMatSubCent * (CenComPorcInst * AuxDcto);

    , {j, 1, AuxNumSubCent }

  ];

  (* asigna valor final*)
  CenComInstRem[[i]] = AuxCostIntR;

  (* elimina aplicacion de factor de instalacion a elementos identificados *)
  (* Equipos de Conservación *)

  CenComInstRem[[i, 1, 3]] = CenComInstRem[[i, 1, 3]] * 0;

  ,
  {i, 1, CenComN}
];
```

■ Total {CenComTotCab, CenComTotRem}

```
(* Costo Total Cabeceras *)
CenComTotCab = CenComMatCab + CenComInstCab ;

(* Costo Total Remotas *)
CenComTotRem = CenComMatRem + CenComInstRem;
```


Distribucion de Costos

■ Funciones de Distribución y Asignacion de Costo

■ Cabecera : Asignación Directa {AsignacionDirectaCabecera}

```

AsignacionDirectaCabecera[mNumPSTN_, mNumRDSIBA_, mNumRDSIPRI_, mPrec_] := Module[
  {mauxDist, i, j, k, m, a, b, c, s},

  (* Crea Matriz Con Resultado Final de asignacion directa*)
  mauxDist = auxDist;

  (* procesa para cada subsistema *)
  (* subsistema de abonados *)
  s = 1;
  a = mPrec[[s]];

  (*TARJETA PARA 8 ABONADOS ACA-3*);
  mauxDist[[s, 1, CenComCatCostoAccesoPSTN]] = a[[1]];
  (*TARJETA KR PARA LSM DE ACA-3*);
  mauxDist[[s, 2, CenComCatCostoAccesoPSTN]] = a[[2]];
  (*LSM ACA-3 PARA 128 AB.POU.*);
  mauxDist[[s, 3, CenComCatCostoAccesoPSTN]] = a[[3]];
  (*LSM PARA 64 ACC 2 B+D INCL.60*);
  mauxDist[[s, 4, CenComCatCostoAccesoBAS]] = a[[4]];
  (*NUEVO LSM-PRA CON ACC.30 B+D*);
  mauxDist[[s, 5, CenComCatCostoAccesoPRI]] = a[[5]];
  (*ALMACEN RPBC (DUPLICADO)*);
  mauxDist[[s, 6, CenComCatCostoAccesoPSTN]] =
    If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0, 0, mNumPSTN / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[6]]];
  mauxDist[[s, 6, CenComCatCostoAccesoBAS]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0,
    0, mNumRDSIBA * 2 / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[6]]];
  mauxDist[[s, 6, CenComCatCostoAccesoPRI]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0,
    0, mNumRDSIPRI * 30 / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[6]]];
  (*ALMACEN SE-PRM (50) 8 AB.GEN.*);
  mauxDist[[s, 7, CenComCatCostoAccesoPSTN]] = a[[7]];
  (*TARJETA JTC PARA RDSI*);
  mauxDist[[s, 8, CenComCatCostoTrafico]] = a[[8]];
  (*ALMACEN STC.*);
  mauxDist[[s, 9, CenComCatCostoTrafico]] = a[[9]];
  (*PREINSTALACION ACA-3*);
  mauxDist[[s, 10, CenComCatCostoAccesoPSTN]] = a[[10]];
  (*PREINSTALACION 2 B+D*);
  mauxDist[[s, 11, CenComCatCostoAccesoBAS]] = a[[11]];
  (*MTU PARA CENTRALES*);
  mauxDist[[s, 12, CenComCatCostoAccesoPSTN]] = a[[12]];
  (*ARMARIO (24 BM) CON 6 ALTURAS.-LSM's*); mauxDist[[s, 13, CenComCatCostoAccesoPSTN]] =
    If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0, 0, mNumPSTN / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[13]]];
  mauxDist[[s, 13, CenComCatCostoAccesoBAS]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0,
    0, mNumRDSIBA * 2 / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[13]]];
  mauxDist[[s, 13, CenComCatCostoAccesoPRI]] = If[(mNumPSTN + mNumRDSIBA + mNumRDSIPRI) == 0,
    0, mNumRDSIPRI * 30 / (mNumPSTN + mNumRDSIBA * 2 + mNumRDSIPRI * 30) * a[[13]]];
  (*ARMARIO (24 BM) CON 6 ALTURAS.-STC's*);
  mauxDist[[s, 14, CenComCatCostoTrafico]] = a[[14]];

  (* subsistema selector de grupo *)
  s = 2;
  a = mPrec[[s]];

  (*ALM.TSM (DUP.) ENTRE 0 Y 32K*);
  mauxDist[[s, 1, CenComCatCostoTrafico]] = a[[1]];
  (*ALMACEN SPM (DUP.) PARA 8K*);
  mauxDist[[s, 2, CenComCatCostoTrafico]] = a[[2]];
  (*ALMACEN RP (DUP.) CON 256 KW.*);
  mauxDist[[s, 3, CenComCatCostoTrafico]] = a[[3]];
  (*ALM.CLM ENTRE 0 Y 32K*);
  mauxDist[[s, 4, CenComCatCostoTrafico]] = a[[4]];
  (*ALMACEN RCM*);
  mauxDist[[s, 5, CenComCatCostoTrafico]] = a[[5]];
  (*ALMACEN ICM*);
  mauxDist[[s, 6, CenComCatCostoTrafico]] = a[[6]];
  (*ALMACEN PCD (DL2)*);
  mauxDist[[s, 7, CenComCatCostoTrafico]] = a[[7]];

```

```

(*ARMARIO 12 BM CON 6 ALTURAS*);
mauxDist[[s, 8, CenComCatCostoTrafico]] = a[[8]];

(* subsistema de enlace y señalizacion *)
s = 3;
a = mPrec[[s]];

(*FUNCIÓN ETC3/4 (32 CANALES)*);
mauxDist[[s, 1, CenComCatCostoTrafico]] = a[[1]];
(*ALMACÉN GEMM1 PARA 16 ETCC4*);
mauxDist[[s, 2, CenComCatCostoTrafico]] = a[[2]];
(*FUNCIÓN TRH (FH,PARA,ST7)*);
mauxDist[[s, 3, CenComCatCostoSenalizacion]] = a[[3]];
(*ALMACÉN TRH (FH,PARA,ST7)*);
mauxDist[[s, 4, CenComCatCostoSenalizacion]] = a[[4]];

(* subsistema de control *)
s = 4;
a = mPrec[[s]];

(*FDD (FLOPPY DISK) CON 1.2 MB*);
mauxDist[[s, 1, CenComCatCostoTrafico]] = a[[1]];
(*HDD (HARD DISK) CON 1 GB*);
mauxDist[[s, 2, CenComCatCostoTrafico]] = a[[2]];
(*TARJETA RPU (INTERFAZ V-24)*);
mauxDist[[s, 3, CenComCatCostoTrafico]] = a[[3]];
(*TARJETA LIU (INTERFAZ V-24)*);
mauxDist[[s, 4, CenComCatCostoTrafico]] = a[[4]];
(*TARJETA ALAMP*);
mauxDist[[s, 5, CenComCatCostoTrafico]] = a[[5]];
(*TARJETA ALEX*);
mauxDist[[s, 6, CenComCatCostoTrafico]] = a[[6]];
(*TARJETA SCAN*);
mauxDist[[s, 7, CenComCatCostoTrafico]] = a[[7]];
(*ALMACÉN ODM*);
mauxDist[[s, 8, CenComCatCostoTrafico]] = a[[8]];
(*APZ 212 20*);
mauxDist[[s, 9, CenComCatCostoTrafico]] = a[[9]];
(*AMPLIACIÓN DE 256 MW (DUP.)*);
mauxDist[[s, 10, CenComCatCostoTrafico]] = a[[10]];
(*IOG 11-B5 BÁSICO*);
mauxDist[[s, 11, CenComCatCostoTrafico]] = a[[11]];

(* otros costos *)
s = 7;
a = mPrec[[s]];

(*SOFTWARE POR ACCESO BASICO*);
mauxDist[[s, 1, CenComCatCostoTrafico]] = a[[1]];
(*SOFTWARE POR ACCESO PRIMARIO*);
mauxDist[[s, 2, CenComCatCostoTrafico]] = a[[2]];
(*DDF*);
mauxDist[[s, 3, CenComCatCostoTrafico]] = a[[3]];

Return[mauxDist];

];

```

■ Remota : Asignación Directa {AsignacionDirectaRemota}

```

(* Esta funcion se aplica para cada subcentral *)
AsignacionDirectaRemota[mPrec_] :=
Module[{mauxDistr, i, j, k, m, a, b, c, s},
  (*Crea Matriz Con Resultado Final de asignacion directa*)
  mauxDistr = auxDistr;

  (*procesa para cada subsistema*)

  (*subsistema de abonados*)
  s = 1;
  a = mPrec[[s]];

  (*TARJETA PARA 8 ABONADOS ACA-3.*);
  mauxDistr[[s, 1, CenComCatCostoAccesoPSTN]] = a[[1]];
  (*TARJETA PARA 4 ACCESOS 2B+D.*);
  mauxDistr[[s, 2, CenComCatCostoAccesoBAS]] = a[[2]];
  (*TARJETA KR PARA LSM DE ACA-3.*);
  mauxDistr[[s, 3, CenComCatCostoAccesoPSTN]] = a[[3]];
  (*LSM ACA-3 128. STR,KR,ETB,POU.*);
  mauxDistr[[s, 4, CenComCatCostoAccesoPSTN]] = a[[4]];
  (*LSM ACA-3 128. KR,ETB,POU.*);
  mauxDistr[[s, 5, CenComCatCostoAccesoPSTN]] = a[[5]];
  (*LSM ACA-3 128. ETB,POU.*);
  mauxDistr[[s, 6, CenComCatCostoAccesoPSTN]] = a[[6]];
  (*LSM ACA-3 128. POU.*);
  mauxDistr[[s, 7, CenComCatCostoAccesoPSTN]] = a[[7]];
  (*LSM ACA-3 PARA 128 AB.POU.*);
  mauxDistr[[s, 8, CenComCatCostoAccesoPSTN]] = a[[8]];
  (*LSM PARA 64 ACC.2B+D.INCLUYE 8 ACCESOS*);
  mauxDistr[[s, 9, CenComCatCostoAccesoBAS]] = a[[9]];
  (*LSM PARA 64 ACC.2B+D.INCLUYE 16 ACCESOS*);
  mauxDistr[[s, 10, CenComCatCostoAccesoBAS]] = a[[10]];
  (*LSM PARA 64 ACC.2B+D.INCLUYE 32 ACCESOS*);
  mauxDistr[[s, 11, CenComCatCostoAccesoBAS]] = a[[11]];
  (*LSM PARA 64 ACC.2B+D.INCLUYE 48 ACCESOS*);
  mauxDistr[[s, 12, CenComCatCostoAccesoBAS]] = a[[12]];
  (*LSM PARA 64 ACC.2B+D.INCLUYE 60 ACCESOS*);
  mauxDistr[[s, 13, CenComCatCostoAccesoBAS]] = a[[13]];
  (*TARJETA ETB PARA R.D.S.I.*);
  mauxDistr[[s, 14, CenComCatCostoTrafico]] = a[[14]];
  (*TARJETA STR PARA LSM DE ACA-3.*);
  mauxDistr[[s, 15, CenComCatCostoTrafico]] = a[[15]];
  (*TARJETA STR PARA R.D.S.I.*);
  mauxDistr[[s, 16, CenComCatCostoTrafico]] = a[[16]];
  (*PREINSTALACION ACA-3.*);
  mauxDistr[[s, 17, CenComCatCostoAccesoPSTN]] = a[[17]];
  (*PREINSTALACION 2B+D.*);
  mauxDistr[[s, 18, CenComCatCostoAccesoBAS]] = a[[18]];

  (*otros costos*)
  s = 4;
  a = mPrec[[s]];

  (*SOFTWARE POR ACCESO BASICO.*);
  mauxDistr[[s, 1, CenComCatCostoTrafico]] = a[[1]];
  (*DDF*);
  mauxDistr[[s, 2, CenComCatCostoTrafico]] = a[[2]];

  Return[mauxDistr];];

```

■ Cabecera : Asignación Indirecta {AsignacionIndirectaCabecera}

```

AsignacionIndirectaCabecera[mPrec_, mAuxDir_] := Module[
  {mauxDist, i, j, k, m, a, b, c, s, auxS, auxP},

  (* Crea Matriz Con Resultado Final de asignacion indirecta*)
  mauxDist = auxDist;

  (* ----- *)
  (* calcula total de inversion por tipo de costo *)
  (* numero de subsistemas *)
  m = Length[mAuxDir];
  (* inversion directa por tipo de costo *)
  
$$\text{auxS} = \sum_{b=1}^m (\text{Plus} @@ \text{mAuxDir} [[b]]);$$


  (* ----- *)
  (* distribucion indirecta en funcion al total de tipo de costo *)
  (* porcentajes de distribucion indirecta *)
  
$$\text{auxP} = \frac{\text{auxS}}{\text{Plus} @@ \text{auxS}};$$

  (* abonado *)
  (*s=1;
  a=mPrec[[s]];
  Do[mauxDist[[s,i],a[[i]]*auxP;,
    {i,{1,3,4,5}}];*)

  (* material y equipo diverso *)
  s = 5;
  a = mPrec[[s]];

  m = Length[a];
  Do[
    mauxDist[[s, i]] = a[[i]] * auxP;
    ,
    {i, 1, m}];

  (* equipamiento de conservacion *)
  s = 6;
  a = mPrec[[s]];

  m = Length[a];
  Do[
    mauxDist[[s, i]] = a[[i]] * auxP;
    ,
    {i, 1, m}];

  (* ----- *)
  (* distribucion indirecta en funcion a tipo de costo Trafico y Señalización *)
  (* porcentajes de distribucion indirecta *)
  m = Length[CenComCatCosto];
  auxP = Table[0, {m}];
  
$$\text{auxP}[[\text{CenComCatCostoTrafico}]] = \frac{\text{auxS}[[\text{CenComCatCostoTrafico}]]}{\text{auxS}[[\text{CenComCatCostoTrafico}]] + \text{auxS}[[\text{CenComCatCostoSenalizacion}]]};$$

  
$$\text{auxP}[[\text{CenComCatCostoSenalizacion}]] = \frac{\text{auxS}[[\text{CenComCatCostoSenalizacion}]]}{\text{auxS}[[\text{CenComCatCostoTrafico}]] + \text{auxS}[[\text{CenComCatCostoSenalizacion}]]};$$


  (* subsistema de enlace y señalizacion *)
  s = 3;
  a = mPrec[[s]];

  (* solo algunos elementos *)
  Do[
    mauxDist[[s, i]] = a[[i]] * auxP;
    ,
    {i, 5, 13}];

  Return[mauxDist];

];

```

Remota : Asignación Indirecta {AsignacionIndirectaRemota}

```

AsignacionIndirectaRemota[mPrec_, mAuxDir_] := Module[
{mauxDistr, i, j, k, m, a, b, c, s, auxS, auxP},

(* Crea Matriz Con Resultado Final de asignacion indirecta*)
mauxDistr = auxDistr;

(* ----- *)
(* calcula total de inversion por tipo de costo *)
(* numero de subsistemas *)
m = Length[mAuxDir];
(* inversion directa por tipo de costo *)

$$\text{auxS} = \sum_{b=1}^m (\text{Plus} @@ \text{mAuxDir} [[b]]);$$


(* ----- *)
(* distribucion indirecta en funcion al total de tipo de costo *)
(* porcentajes de distribucion indirecta *)
If[(Plus @@ auxS) == 0,
  (* inversion = 0, los factores de distribucion no existen matematicamente *)
  auxP = Table[0, {i, Length[CenComCatCosto]}];
,
  (* inversion <> 0. si existen factores de distribucion matematicamente *)
  auxP = auxS / Plus @@ auxS;
];

(* Sistema de Abonado *)
(* solo algunos elementos *)
s = 1;
a = mPrec[[s]];
Do[
  mauxDistr[[s, i]] = a[[i]] * auxP;
,
  {i, 19, 20}];

(* material y equipo diverso *)
s = 2;
a = mPrec[[s]];

m = Length[a];
Do[
  mauxDistr[[s, i]] = a[[i]] * auxP;
,
  {i, 1, m}];

(* equipamiento de conservacion *)
s = 3;
a = mPrec[[s]];

m = Length[a];
Do[
  mauxDistr[[s, i]] = a[[i]] * auxP;
,
  {i, 1, m}];

Return[mauxDistr];

];

```

■ **Asignación Directa e Indirecta {CenComDistDirCab, CenComDistIndirCab, CenComDistCab, CenComDistDirRem, CenComDistIndirRem, CenComDistRem}**

■ **Creacion de matrices de variables resultado**

```
(* para Cabeceras *)
CenComDistDirCab = Table[auxDist, {i, 1, CenComN}];
CenComDistIndirCab = Table[auxDist, {i, 1, CenComN}];
CenComDistCab = Table[auxDist, {i, 1, CenComN}];
(* para Unidades Remotas *)
CenComDistDirRem = Table[{auxDistr}, {i, 1, CenComN}];
CenComDistIndirRem = Table[{auxDistr}, {i, 1, CenComN}];
CenComDistRem = Table[{auxDistr}, {i, 1, CenComN}];
```

■ **Calculo y compilación de Asignación Directa e Indirecta**

```
Do[
  NumPSTN = CenComInput[[i, 6]];
  NumRDSIBA = CenComInput[[i, 7]];
  NumRDSIPRI = CenComInput[[i, 8]];
  xAuxTot = CenComTotCab[[i]];
  xAuxTotr = CenComTotRem[[i]];

  If[ CenComInput[[i, 5]] == "CABECERA",
    CenComDistDirCab[[i]] = AsignacionDirectaCabecera[NumPSTN, NumRDSIBA, NumRDSIPRI, xAuxTot];
    xAuxDir = CenComDistDirCab[[i]];
    CenComDistIndirCab[[i]] = AsignacionIndirectaCabecera[xAuxTot, xAuxDir];,
    m = Length[xAuxTotr];
    auxR = Table[AsignacionDirectaRemota[xAuxTotr[[j]]], {j, 1, m}];
    CenComDistDirRem[[i]] = auxR;
    auxR = Table[AsignacionIndirectaRemota[xAuxTotr[[j]], auxR[[j]]], {j, 1, m}];
    CenComDistIndirRem[[i]] = auxR;];
  , {i, 1, CenComN}];

(* distribucion total *)
CenComDistCab = CenComDistDirCab + CenComDistIndirCab;
CenComDistRem = CenComDistDirRem + CenComDistIndirRem;
```

Resultado Final

■ **Creacion de Vectores de Tipo de Costo "Trafico" {CenComTraficoCab, CenComTraficoRem}**

```
CenComTraficoCab = Table[0, {i, 1, CenComN}];
CenComTraficoRem = Table[0, {i, 1, CenComN}];
```

■ **Funciones Auxiliares**

■ **Funcion que extrae la inversion en una categoria de costo especifica {fCatCostCent}**

$$fCatCostCent[mCentral_, mCatCost_] := \sum_{ms=1}^{Length[mCentral]} \left(\sum_{me=1}^{Length[mCentral][[ms]]} mCentral[[ms, me, mCatCost]] \right);$$

■ Extraccion de la Inversion en Trafico {CenComTraficoCab, CenComTraficoRem}

```

CenComTraficoCab = Table[
  fCatCostCent[CenComDistCab[[x]], CenComCatCostoTrafico]
  ,
  {x, 1, Length[CenComDistCab]}
];

CenComTraficoRem = Table[
  Plus @@ (
    Table[
      fCatCostCent[CenComDistRem[[x, y]], CenComCatCostoTrafico]
      ,
      {y, 1, Length[ CenComDistRem[[x]] ]}
    ]
  )
  ,
  {x, 1, Length[CenComDistRem]}
];

```

■ Total {CenComTrafico}

```

CenComTrafico = CenComTraficoCab + CenComTraficoRem;

```

■ Guarda el Archivo en disco

```

ArchSalCommInvTraf = "Switchinv.txt";
Export[ArchSalCommInvTraf, CenComTrafico, "CSV"];

```

■ Impresion de la inversion en Trafico Total (Cabeceras y Unidades Remotas)

```

Module[
  {aT, aTh, aR},
  aT = Transpose[CenComInput] // TableForm;
  aTh = {None, {"CentralId", "Departamento", "Provincia", "Central", "Inversion Conmutacion"}};
  aR = Transpose[Table[{aT[[1, 1]], aT[[1, 2]], aT[[1, 3]], aT[[1, 4]], CenComTrafico}]];
  (* *)
  Print[TableForm[aR, TableHeadings → aTh]];

];

```

INVERSIÓN EN CONMUTACIÓN A NIVEL DE CENTRALES LOCALES

```

Print[Style["INVERSIONES EN CONMUTACIÓN A NIVEL DE CENTRALES LOCALES",
  Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["CVoz"];

CVoz = Ceiling[ $\frac{\text{trunkx}}{30}$  / fiberfill];

(* Se calcula los circuitos (E1s) de voz por cada central *)
Print["CosteAñadidoPorDistanciax"];
CosteAñadidoPorDistanciax = zerox;
Do[If[deptx[[ix]] ≠ LIMA,
  CosteAñadidoPorDistanciax[[ix]] = AñadidoCxPorKm * distance[ix, idinvertx[[CentralRef]]];], {ix, nx}];

(* Se Calcula el Coste Añadido por distancia para cada central fuera de Lima *)
Print[TableForm[Table[{ix, y[[ix, 13]], CVoz[[ix]], CosteAñadidoPorDistanciax[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "CVoz", "CosteAñadidoPorDistanciax"}}]];

OpenRead["Switchinv.txt"];
switchinvx = zerox;
switchinvLocx = zerox;
switchinvLDNx = zerox;
switchinvLDIx = zerox;
switchinvx = ReadList["Switchinv.txt", Number];
switchinvx = switchinvx + CosteAñadidoPorDistanciax;

Do[If[ix ≠ idinvertx[[248]],
  SwITot = If[switchendLocx[[ix]] ≥ 0, switchendLocx[[ix]], 0] +
    If[switchendLDNx[[ix]] ≥ 0, switchendLDNx[[ix]], 0] + If[switchendLDIx[[ix]] ≥ 0, switchendLDIx[[ix]], 0];
  If[SwITot ≠ 0, switchinvLocx[[ix]] = If[switchendLocx[[ix]] ≥ 0, switchendLocx[[ix]], 0] *  $\frac{\text{switchinvx}[[ix]]}{\text{SwITot}}$ ;

  switchinvLDNx[[ix]] = If[switchendLDNx[[ix]] ≥ 0, switchendLDNx[[ix]], 0] *  $\frac{\text{switchinvx}[[ix]]}{\text{SwITot}}$ ;

  switchinvLDIx[[ix]] = If[switchendLDIx[[ix]] ≥ 0, switchendLDIx[[ix]], 0] *  $\frac{\text{switchinvx}[[ix]]}{\text{SwITot}}$ ];], {ix, nx}];

Print[TableForm[Table[{ix, y[[ix, 13]], switchinvx[[ix]], switchinvLocx[[ix]], switchinvLDNx[[ix]], switchinvLDIx[[ix]]},
  {ix, nx}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Nombre Central", "switchinvx", "switchinvLocx", "switchinvLDNx", "switchinvLDIx"}}]];

(* Se lee el archivo que contiene las inversiones ya en costo de conmutación *)

switchinvd = zerod;
switchinvLocd = zerod;
switchinvLDNd = zerod;
switchinvLDId = zerod;
Do[switchinvd[[deptx[[ix]]]] = switchinvd[[deptx[[ix]]]] + switchinvx[[ix]];
  switchinvLocd[[deptx[[ix]]]] = switchinvLocd[[deptx[[ix]]]] + switchinvLocx[[ix]];
  switchinvLDNd[[deptx[[ix]]]] = switchinvLDNd[[deptx[[ix]]]] + switchinvLDNx[[ix]];
  switchinvLDId[[deptx[[ix]]]] = switchinvLDId[[deptx[[ix]]]] + switchinvLDIx[[ix]]], {ix, nx}];

Print[TableForm[
  Table[{id, deptTXTd[id]], switchinvd[id], switchinvLocd[id], switchinvLDNd[id], switchinvLDId[id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Departamento", "switchinvd", "switchinvLocd", "switchinvLDNd", "switchinvLDId"}}]];

(* Se acumula la inversión a nivel de departamento *)

hostsatinvx = host1x * sat1x * switchinvx;
hostsatinvLocx = host1x * sat1x * switchinvLocx;
hostsatinvLDNx = host1x * sat1x * switchinvLDNx;
hostsatinvLDIx = host1x * sat1x * switchinvLDIx;
Print[TableForm[
  Table[{ix, y[[ix, 13]], hostsatinvx[[ix]], hostsatinvLocx[[ix]], hostsatinvLDNx[[ix]], hostsatinvLDIx[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Nombre Central", "hostsatinvx", "hostsatinvLocx", "hostsatinvLDNx", "hostsatinvLDIx"}}]];

(* Cabeceras Satelitales *)

hostsatinvd = zerod;
hostsatinvLocd = zerod;
hostsatinvLDNd = zerod;
hostsatinvLDId = zerod;
Do[hostsatinvd[[deptx[[ix]]]] = hostsatinvd[[deptx[[ix]]]] + hostsatinvx[[ix]];
  hostsatinvLocd[[deptx[[ix]]]] = hostsatinvLocd[[deptx[[ix]]]] + hostsatinvLocx[[ix]];
  hostsatinvLDNd[[deptx[[ix]]]] = hostsatinvLDNd[[deptx[[ix]]]] + hostsatinvLDNx[[ix]];
  hostsatinvLDId[[deptx[[ix]]]] = hostsatinvLDId[[deptx[[ix]]]] + hostsatinvLDIx[[ix]]], {ix, nx}];

Print[
  TableForm[Table[{id, deptTXTd[id]], hostsatinvd[id], hostsatinvLocd[id], hostsatinvLDNd[id], hostsatinvLDId[id]],

```

```

{id, nd}}, TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
{None, {"id", "Departamento", "hostsatinvd", "hostsatinvLocd", "hostsatinvLDNd", "hostsatinvLDId"}}];
(* Se acumulan las inversiones de cabeceras satelitales a nivel de departamento *)

hostsatinv = Plus @@ hostsatinvd;
hostsatinvLoc = Plus @@ hostsatinvLocd;
hostsatinvLDN = Plus @@ hostsatinvLDNd;
hostsatinvLDI = Plus @@ hostsatinvLDId;
Print[TableForm[{hostsatinv, hostsatinvLoc, hostsatinvLDN, hostsatinvLDI},
  TableDirections → {Row, Column}, TableSpacing → {0, 1},
  TableHeadings → {"hostsatinv", "hostsatinvLoc", "hostsatinvLDN", "hostsatinvLDI"}, None]];
(* Se acumulan las inversiones de cabeceras satelitales a nivel nacional *)

tandeminvx = 2 * lima1x * tandem1x * switchinvx;
tandeminvLocx = 2 * lima1x * tandem1x * switchinvLocx;
tandeminvLDNx = 2 * lima1x * tandem1x * switchinvLDNx;
tandeminvLDIx = 2 * lima1x * tandem1x * switchinvLDIx;
Print[TableForm[Table[{ix, y[ix, 13], tandeminvx[ix], tandeminvLocx[ix], tandeminvLDNx[ix], tandeminvLDIx[ix]},
  {ix, nx}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "Nombre Central", "tandeminvx", "tandeminvLocx", "tandeminvLDNx", "tandeminvLDIx"}}]];
(* Inversión en Tándems *)

tandeminvd = zerod;
tandeminvLocd = zerod;
tandeminvLDNd = zerod;
tandeminvLDId = zerod;
Do[tandeminvd[depts[ix]] = tandeminvd[depts[ix]] + tandeminvx[ix];
  tandeminvLocd[depts[ix]] = tandeminvLocd[depts[ix]] + tandeminvLocx[ix];
  tandeminvLDNd[depts[ix]] = tandeminvLDNd[depts[ix]] + tandeminvLDNx[ix];
  tandeminvLDId[depts[ix]] = tandeminvLDId[depts[ix]] + tandeminvLDIx[ix];, {ix, nx}];
Print[TableForm[
  Table[{id, deptTXTd[id], tandeminvd[id], tandeminvLocd[id], tandeminvLDNd[id], tandeminvLDId[id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Departamento", "tandeminvd", "tandeminvLocd", "tandeminvLDNd", "tandeminvLDId"}}]];
(*tándem de Lima es doble*)

hostnonsatinvx = host1x * switchinvx - hostsatinvx -  $\frac{\text{tandeminvx}}{2}$ ;
hostnonsatinvLocx = host1x * switchinvLocx - hostsatinvLocx -  $\frac{\text{tandeminvLocx}}{2}$ ;
hostnonsatinvLDNx = host1x * switchinvLDNx - hostsatinvLDNx -  $\frac{\text{tandeminvLDNx}}{2}$ ;
hostnonsatinvLDIx = host1x * switchinvLDIx - hostsatinvLDIx -  $\frac{\text{tandeminvLDIx}}{2}$ ;
Print[TableForm[
  Table[{ix, y[ix, 13], hostnonsatinvx[ix], hostnonsatinvLocx[ix], hostnonsatinvLDNx[ix], hostnonsatinvLDIx[ix]},
  {ix, nx}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings → {None,
  {"ix", "Nombre Central", "hostnonsatinvx", "hostnonsatinvLocx", "hostnonsatinvLDNx", "hostnonsatinvLDIx"}}]];
(* cabeceras no satelitaleso *)

hostnonsatinvd = zerod;
hostnonsatinvLocd = zerod;
hostnonsatinvLDNd = zerod;
hostnonsatinvLDId = zerod;
Do[hostnonsatinvd[depts[ix]] = hostnonsatinvd[depts[ix]] + hostnonsatinvx[ix];
  hostnonsatinvLocd[depts[ix]] = hostnonsatinvLocd[depts[ix]] + hostnonsatinvLocx[ix];
  hostnonsatinvLDNd[depts[ix]] = hostnonsatinvLDNd[depts[ix]] + hostnonsatinvLDNx[ix];
  hostnonsatinvLDId[depts[ix]] = hostnonsatinvLDId[depts[ix]] + hostnonsatinvLDIx[ix];, {ix, nx}];
Print[TableForm[Table[{id, deptTXTd[id], hostnonsatinvd[id], hostnonsatinvLocd[id], hostnonsatinvLDNd[id],
  hostnonsatinvLDId[id]}, {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings → {None,
  {"id", "Departamento", "hostnonsatinvd", "hostnonsatinvLocd", "hostnonsatinvLDNd", "hostnonsatinvLDId"}}]];
(* Se acumulan las inversiones de cabeceras no satelitales a nivel de departamento *)

hostnonsatinv = Plus @@ hostnonsatinvd;
hostnonsatinvLoc = Plus @@ hostnonsatinvLocd;
hostnonsatinvLDN = Plus @@ hostnonsatinvLDNd;
hostnonsatinvLDI = Plus @@ hostnonsatinvLDId;
Print[TableForm[{hostnonsatinv, hostnonsatinvLoc, hostnonsatinvLDN, hostnonsatinvLDI},
  TableDirections → {Row, Column}, TableSpacing → {0, 1},
  TableHeadings → {"hostnonsatinv", "hostnonsatinvLoc", "hostnonsatinvLDN", "hostnonsatinvLDI"}, None]];
(* Se acumulan las inversiones de cabeceras no satelitales a nivel nacional *)

Print["tandeminvitxd"]
tandeminvitxd = zerod;
Do[tandeminvitxd[depts[ix]] =
  tandeminvitxd[depts[ix]] + tandem1x[ix] * switchinvx[ix] *  $\frac{\text{CITX[tandemx[ix]]}}{\text{CVoz[ix]} + \text{CITX[tandemx[ix]]}}$ ;, {ix, nx}];
tandeminvitxd[LIMA] = 2 tandeminvitxd[LIMA];

```

```

Print[TableForm[Table[{id, deptTXTd[[id]], tandeminvitxd[[id]]}, {id, nd}], , TableDirections → {Column, Row},
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Departamento", "tandeminvitxd"}}]]
(* Se halla la participación de los Circuitos de Interconexión a nivel departamental *)

Print["tandeminvitx"];
tandeminvitx = Plus @@ tandeminvitxd;
Print[tandeminvitx];
(* Se halla la participación Nacional de los Circuitos de Interconexión *)

remoteinvd = switchinvd - hostsatinvd - hostnonsatinvd -  $\frac{\text{tandeminvd}}{2}$ ;
remoteinvLocd = switchinvLocd - hostsatinvLocd - hostnonsatinvLocd -  $\frac{\text{tandeminvLocd}}{2}$ ;
remoteinvLDNd = switchinvLDNd - hostsatinvLDNd - hostnonsatinvLDNd -  $\frac{\text{tandeminvLDNd}}{2}$ ;
remoteinvLDId = switchinvLDId - hostsatinvLDId - hostnonsatinvLDId -  $\frac{\text{tandeminvLDId}}{2}$ ;

Print[TableForm[
  Table[{id, deptTXTd[[id]], remoteinvd[[id]], remoteinvLocd[[id]], remoteinvLDNd[[id]], remoteinvLDId[[id]]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"id", "Departamento", "remoteinvd", "remoteinvLocd", "remoteinvLDNd", "remoteinvLDId"}}]];
(* Participación en Remota por Departamento *)

remoteinv = Plus @@ remoteinvd;
remoteinvLoc = Plus @@ remoteinvLocd;
remoteinvLDN = Plus @@ remoteinvLDNd;
remoteinvLDI = Plus @@ remoteinvLDId;
Print[TableForm[{remoteinv, remoteinvLoc, remoteinvLDN, remoteinvLDI}, TableDirections → {Row, Column},
  TableSpacing → {0, 1}, TableHeadings → {"remoteinv", "remoteinvLoc", "remoteinvLDN", "remoteinvLDI"}, None]]];
(* Participación de Remotas a Nivel Nacional *)

Print["signalinvd"];
signalinvd = signalfrac * switchinvd;
Print[TableForm[Table[{id, deptTXTd[[id]], signalinvd[[id]]}, {id, nd}], TableDirections → {Column, Row},
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Departamento", "signalinvd"}}]];
(* Participación de Señalización a Nivel Departamental*)

Print["signalinv"];
signalinv = Plus @@ signalinvd;
Print[signalinv];
(* Participación de Señalización a Nivel Nacional*)

```

```

Print[Style["TERMINACIONES DE FIBRA", 16, White, Bold, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
Print["costeTerminacionesFibrax"]; Do[If[transtechx[[ix]] == Urbana ∨ transtechx[[ix]] == Enterrado,
  costeTerminacionesFibrax[[ix]] = costeTerminacionesFibrax[[ix]] + 2 TxFibraCosteTerminacion, 0], {ix, nx}];
Print[TableForm[Table[{ix, y[[ix, 13]], costeTerminacionesFibrax[[ix]]}, {ix, nx}], TableSpacing → {0, 1},
  TableHeadings → {None, {"ix", "Nombre Central", "costeTerminacionesFibrax"}}]];
(* En cada enlace de Fibra: En cada extremo de una arista hay dos terminaciones de fibra,
sin embargo, si hay en un anillo es necesario también considerar las
terminaciones de fibra de la arista entre el último nodo y la cabecera del anillo,
variable que se estimó al inicio del modelo en la primera versión de la variable costeTerminacionesFibrax*)

```

DESAGREGACION DE ANILLOS POR ARISTA

```
Print[Style["DESAGREGACIÓN DEL TAMAÑO DE LOS ANILLOS: POR ARISTA",
White, Bold, 16, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print[Style["A NIVEL PROVINCIAL", White, Bold, 16, Underlined, Background → Blue]],

Print["disthostx modificado"];
Do[If[esAnillopx[[ix]] == 1,
If[e1Anillop[[provx[[ix]]]] ≠ 0, disthostx[[ix]] = TamAnillop[[provx[[ix]]]] *  $\frac{e1remotehostTOTALx[[ix]]}{e1AnilloTOTALp[[provx[[ix]]]]}$ ]]; {ix, nx}];
Print[TableForm[Table[{ix, y[[ix, 13]], disthostx[[ix]]}, {ix, nx}], TableSpacings → {1, 0},
TableHeadings → {None, {"ix", "Nombre Central", "disthostx"}}]];
(* Para todas las centrales remotas que forman parte de un anillo provincial,
se modifica la distancia a ser considerada como relevante para luego calcular los
costos de cada arista. Para ello, asigna a cada tramo una fracción del total del tamaño
del anillo. Dicho porcentaje o fracción es equivalente al total de E1s identificados en
el nivel RH (e1remotehostTOTALx) para cada central entre el total de E1s del anillo *)
Print[Style["A NIVEL DEPARTAMENTAL", White, Bold, 16, Underlined, Background → Blue]],

Print["disttandemx modificado"]; Do[If[esAnillodx[[AnilloCabeceras[[ix]]]] ≠ NA,
If[e1Anillod[[deptx[[AnilloCabeceras[[ix]]]]]] ≠ 0, disttandemx[[AnilloCabeceras[[ix]]]] =
TamAnillod[[deptx[[AnilloCabeceras[[ix]]]]]] *  $\frac{e1hosttandemTOTALx[[AnilloCabeceras[[ix]]]]}{e1AnilloTOTALd[[deptx[[AnilloCabeceras[[ix]]]]]]}$ ;]]; {ix, np}];
Print[TableForm[Table[{ix, y[[ix, 13]], disttandemx[[ix]]}, {ix, nx}], TableSpacings → {1, 0},
TableHeadings → {None, {"ix", "Nombre Central", "disttandemx"}}]];
(* Para todas las centrales cabeceras que forman parte de un anillo departamental,
se modifica la distancia a ser considerada como relevante para luego calcular los
costos de cada arista. Para ello, asigna a cada tramo una fracción del total del tamaño
del anillo. Dicho porcentaje o fracción es equivalente al total de E1s identificados en
el nivel HT (e1hosttandemTOTALx) para cada central entre el total de E1s del anillo *)
```

ESTIMACION DE INVERSIONES A NIVEL SATELITAL

```

Print[Style["ESTIMACIÓN DE INVERSIONES A NIVEL SATELITAL",
  16, White, Bold, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print[Style["A NIVEL DE EQUIPOS EN LOS LUGARES REMOTOS",
  16, White, Bold, Background → Blue]];

satinvx = Table[
  If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0, (satantena + sateqtx) *  $\left(\frac{e1hostttandemx[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvLocx = satinvx * satLocx;
satinvLDNx = satinvx * satLDNx;
satinvLDIx = satinvx * satLDIx;
satinvlqxAl = Table[
  If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0, (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxAl[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxB1 = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxB1[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxC1 = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxC1[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxLL = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxLL[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxLP = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxLP[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxAS = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxAS[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxBS = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxBS[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxCS = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxCS[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxABC2 = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxABC2[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvlqxABCnop = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemalqxABCnop[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvitxx = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemitxx[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvadslx = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0,
  (satantena + sateqtx) *  $\left(\frac{e1hostttandemadslx[[ix]]}{e1hostttandemTOTALx[[ix]]}\right)$ ], {ix, nx}];

satinvTOTALx = Table[If[sat1x[[ix]] == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, 0, (satantena + sateqtx)], {ix, nx}];
Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], satinvx[[ix]], satinvLocx[[ix]], satinvLDNx[[ix]], satinvLDIx[[ix]],
  satinvlqxAl[[ix]], satinvlqxB1[[ix]], satinvlqxC1[[ix]], satinvlqxLL[[ix]], satinvlqxLP[[ix]],
  satinvlqxAS[[ix]], satinvlqxBS[[ix]], satinvlqxCS[[ix]], satinvlqxABC2[[ix]], satinvlqxABCnop[[ix]],
  satinvitxx[[ix]], satinvadslx[[ix]], satinvTOTALx[[ix]]}, {ix, nx}], TableSpacing → {0, 1}, TableHeadings →
  {None, {"ix", "idx", "Nombre Central", "satinvx", "satinvLocx", "satinvLDNx", "satinvLDIx", "satinvlqxAl",
    "satinvlqxB1", "satinvlqxC1", "satinvlqxLL", "satinvlqxLP", "satinvlqxAS", "satinvlqxBS",
    "satinvlqxCS", "satinvlqxABC2", "satinvlqxABCnop", "satinvitxx", "satinvadslx", "satinvTOTALx"}}]];

(* Para todas las centrales que utilizan Tx satelital, se estima la inversión en los equipos
  de Tx y la antena satelital. Además, para imputar el costo de cada servicio,
  se multiplica la inversión total por la fracción de asignación,
  la cual es equivalente al ratio entre el total de E1s a nivel HT de
  cada servicio entre el total de E1s a nivel HT que incluye todos los servicios *)

satinvd = zerod;
satinvLocd = zerod;
satinvLDNd = zerod;
satinvLDId = zerod;
satinvlqda1 = zerod;

```




```

satinvalqdB1 = zerod;
satinvalqdC1 = zerod;
satinvalqdLL = zerod;
satinvalqdLP = zerod;
satinvalqdAS = zerod;
satinvalqdBS = zerod;
satinvalqdCS = zerod;
satinvalqdABC2 = zerod;
satinvalqdABCnop = zerod;
satinvadsld = zerod;
satinvitxd = zerod;
satinvTOTALd = zerod;
Do[satinvd[[deptx[ix]]] = satinvd[[deptx[ix]]] + satinvx[ix];
  satinvLocd[[deptx[ix]]] = satinvLocd[[deptx[ix]]] + satinvLocx[ix];
  satinvLDNd[[deptx[ix]]] = satinvLDNd[[deptx[ix]]] + satinvLDNx[ix];
  satinvLDId[[deptx[ix]]] = satinvLDId[[deptx[ix]]] + satinvLDIx[ix];
  satinvalqdA1[[deptx[ix]]] = satinvalqdA1[[deptx[ix]]] + satinvalqxA1[ix];
  satinvalqdB1[[deptx[ix]]] = satinvalqdB1[[deptx[ix]]] + satinvalqxB1[ix];
  satinvalqdC1[[deptx[ix]]] = satinvalqdC1[[deptx[ix]]] + satinvalqxC1[ix];
  satinvalqdLL[[deptx[ix]]] = satinvalqdLL[[deptx[ix]]] + satinvalqxLL[ix];
  satinvalqdLP[[deptx[ix]]] = satinvalqdLP[[deptx[ix]]] + satinvalqxLP[ix];
  satinvalqdAS[[deptx[ix]]] = satinvalqdAS[[deptx[ix]]] + satinvalqxAS[ix];
  satinvalqdBS[[deptx[ix]]] = satinvalqdBS[[deptx[ix]]] + satinvalqxBS[ix];
  satinvalqdCS[[deptx[ix]]] = satinvalqdCS[[deptx[ix]]] + satinvalqxCS[ix];
  satinvalqdABC2[[deptx[ix]]] = satinvalqdABC2[[deptx[ix]]] + satinvalqxABC2[ix];
  satinvalqdABCnop[[deptx[ix]]] = satinvalqdABCnop[[deptx[ix]]] + satinvalqxABCnop[ix];
  satinvadsld[[deptx[ix]]] = satinvadsld[[deptx[ix]]] + satinvadslx[ix];
  satinvitxd[[deptx[ix]]] = satinvitxd[[deptx[ix]]] + satinvitxx[ix];
  satinvTOTALd[[deptx[ix]]] = satinvTOTALd[[deptx[ix]]] + satinvTOTALx[ix];
, {ix, nx}];
Print[
  TableForm[Table[{id, deptTXTd[id], satinvd[id], satinvLocd[id], satinvLDNd[id], satinvLDId[id], satinvalqdA1[id],
    satinvalqdB1[id], satinvalqdC1[id], satinvalqdLL[id], satinvalqdLP[id], satinvalqdAS[id], satinvalqdBS[id],
    satinvalqdCS[id], satinvalqdABC2[id], satinvalqdABCnop[id], satinvadsld[id],
    satinvitxd[id], satinvTOTALd[id]}, {id, nd}], TableSpacing -> {0, 1}, TableHeadings ->
    {None, {"id", "Nombre Departamento", "satinvd", "satinvLocd", "satinvLDNd", "satinvLDId", "satinvalqdA1",
      "satinvalqdB1", "satinvalqdC1", "satinvalqdLL", "satinvalqdLP", "satinvalqdAS", "satinvalqdBS",
      "satinvalqdCS", "satinvalqdABC2", "satinvalqdABCnop", "satinvadsld", "satinvitxd", "satinvTOTALd "}}]];
(* Acumula a nivel departamental las variables satinvx, satinvLocx, satinvLDNx, satinvLDIx,
satinvalqxA1, satinvalqxB1, satinvalqxC1, satinvalqxLL, satinvalqxLP, satinvalqxAS,
satinvalqxBS, satinvalqxCS, satinvalqxA1nop, satinvalqxB1nop, satinvadslx,
satinvitxx: total de inversión en equipos de Tx y antenas satelitales de cada servicio *)

satinv = Plus @@ satinvd;
satinvLoc = Plus @@ satinvLocd;
satinvLDN = Plus @@ satinvLDNd;
satinvLDI = Plus @@ satinvLDId;
satinvalqA1 = Plus @@ satinvalqdA1;
satinvalqB1 = Plus @@ satinvalqdB1;
satinvalqC1 = Plus @@ satinvalqdC1;
satinvalqLL = Plus @@ satinvalqdLL;
satinvalqLP = Plus @@ satinvalqdLP;
satinvalqAS = Plus @@ satinvalqdAS;
satinvalqBS = Plus @@ satinvalqdBS;
satinvalqCS = Plus @@ satinvalqdCS;
satinvalqABC2 = Plus @@ satinvalqdABC2;
satinvalqABCnop = Plus @@ satinvalqdABCnop;
satinvadsld = Plus @@ satinvadsld;
satinvitx = Plus @@ satinvitxd;
satinvTOTAL = Plus @@ satinvTOTALx;
Print[TableForm[{satinv, satinvLoc, satinvLDN, satinvLDI, satinvalqA1, satinvalqB1,
  satinvalqC1, satinvalqLL, satinvalqLP, satinvalqAS, satinvalqBS, satinvalqCS, satinvalqABC2,
  satinvalqABCnop, satinvadsl, satinvitx, satinvTOTAL}, TableDirections -> {Row, Column},
  TableHeadings -> {{{"satinv", "satinvLoc", "satinvLDN", "satinvLDI", "satinvalqA1", "satinvalqB1",
    "satinvalqC1", "satinvalqLL", "satinvalqLP", "satinvalqAS", "satinvalqBS", "satinvalqCS",
    "satinvalqABC2", "satinvalqABCnop", "satinvadsld", "satinvitx", "satinvTOTAL"}, Automatic}}]];
(*Sumatoria a nivel nacional del total de inversión en equipos de Tx y antenas satelitales de cada servicio *)

```

```

Print[Style["ESTIMACIÓN DE INVERSIONES A NIVEL SATELITAL",
  16, White, Bold, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];
Print[Style["A NIVEL DE EQUIPOS EN LIMA", 16, White, Bold, Background -> Blue]];
Print["numerosat"];
numerosat = 0;
Do[If[sat1x[ix] == 1, numerosat = numerosat + 1], {ix, nx}];
Print[numerosat];
(* Identifica a nivel nacional el número total de centrales que emplean Tx satelital *)

```

```

Print["satlimainvx"];
satlimainvx = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvx[[ix]] = 0,
    satlimainvx[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemx}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];
Print[satlimainvx];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Voz,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de
Voz entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvLocx"];
satlimainvLocx = satlimainvx * satLocx;
Print[satlimainvLocx];
(* Parte de Tráfico Local de Voz *)

Print["satlimainvLDNx"];
satlimainvLDNx = satlimainvx * satLDNx;
Print[satlimainvLDNx];
(* Parte de Tráfico LDN de Voz *)

Print["satlimainvLDIx"];
satlimainvLDIx = satlimainvx * satLDIx;
Print[satlimainvLDIx];
(* Parte de Tráfico LDI de Voz *)

Print["satlimainvalqxA1"];
satlimainvalqxA1 = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxA1[[ix]] = 0,
    satlimainvalqxA1[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqx}A1[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];
Print[satlimainvalqxA1];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos Rango A,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos
Rango A entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxB1"];
satlimainvalqxB1 = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxB1[[ix]] = 0,
    satlimainvalqxB1[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqx}B1[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];
Print[satlimainvalqxB1];
(* Dedido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos Rango B,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos
Rango B entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxC1"];
satlimainvalqxC1 = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxC1[[ix]] = 0,
    satlimainvalqxC1[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqx}C1[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];
Print[satlimainvalqxC1];
(* Dedido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos Rango C,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos
Rango B entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxLL"];

```

```

satlimainvalqxLL = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxLL[[ix]] = 0,

    satlimainvalqxLL[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqxLL}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];

Print[satlimainvalqxLL];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos Local Lima,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos
Local Lima entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxLP"];
satlimainvalqxLP = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxLP[[ix]] = 0,

    satlimainvalqxLP[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqxLP}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];

Print[satlimainvalqxLP];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además,
para imputar el costo a Circuitos Local Provincia,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos Local
Provincia entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxAS"];
satlimainvalqxAS = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxAS[[ix]] = 0,

    satlimainvalqxAS[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqxAS}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];

Print[satlimainvalqxAS];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además,
para imputar el costo a Circuitos Rango A Satelite,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos Rango
A Satelital entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxBS"];
satlimainvalqxBS = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxBS[[ix]] = 0,

    satlimainvalqxBS[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqxBS}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];

Print[satlimainvalqxBS];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además,
para imputar el costo a Circuitos Rango B Satelital,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos Rango
B Satelital entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

Print["satlimainvalqxCS"];
satlimainvalqxCS = zerox;
Do[If[numerosat == 0 ∨ e1hosttandemTOTALx[[ix]] == 0, satlimainvalqxCS[[ix]] = 0,

    satlimainvalqxCS[[ix]] =  $\left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx}\right) * \left(\frac{\text{e1hosttandemalqxCS}[[ix]]}{\text{e1hosttandemTOTALx}[[ix]]}\right) * \text{sat1x}[[ix]]$ , {ix, nx}];

Print[satlimainvalqxCS];
(* Debido a que todas las transmisiones satelitales vienen hasta Lima,
en Lima se requiere también un equipo de Tx y una antena satelital. Entonces,
Si existe por lo menos una central que emplee Tx satelital se hace lo
siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima,
y además una fracción proporcional del costo de la antena. Además,
para imputar el costo a Circuitos Rango C Satelital,
se multiplica la inversión adicional estimada por la fracción de asignación,
la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos Rango

```

C Satelital entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

```
Print["satlimainvalqxABC2"];
```

```
satlimainvalqxABC2 = zerox;
```

```
Do[If[numerosat == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, satlimainvalqxABC2[[ix]] = 0,
```

$$\text{satlimainvalqxABC2}[[ix]] = \left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx} \right) * \left(\frac{\text{e1hostttandemalqxABC2}[[ix]]}{\text{e1hostttandemTOTALx}[[ix]]} \right) * \text{sat1x}[[ix]], \{ix, nx\}];$$

```
Print[satlimainvalqxABC2];
```

(* Debido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo de Tx y una antena satelital. Entonces, Si existe por lo menos una central que emplee Tx satelital se hace lo siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima, y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Otros Circuitos Rango A, B y C, se multiplica la inversión adicional estimada por la fracción de asignación, la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Otros Circuitos Rango A, B y C entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

```
Print["satlimainvalqxABCnop"];
```

```
satlimainvalqxABCnop = zerox;
```

```
Do[If[numerosat == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, satlimainvalqxABCnop[[ix]] = 0,
```

$$\text{satlimainvalqxABCnop}[[ix]] = \left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx} \right) * \left(\frac{\text{e1hostttandemalqxABCnop}[[ix]]}{\text{e1hostttandemTOTALx}[[ix]]} \right) * \text{sat1x}[[ix]], \{ix, nx\}];$$

```
Print[satlimainvalqxABCnop];
```

(* Dedido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo de Tx y una antena satelital. Entonces, Si existe por lo menos una central que emplee Tx satelital se hace lo siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima, y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos de No Operadores, se multiplica la inversión adicional estimada por la fracción de asignación, la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos de No Operadores entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

```
Print["satlimainvads1x"];
```

```
satlimainvads1x = zerox;
```

```
Do[If[numerosat == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, satlimainvads1x[[ix]] = 0,
```

$$\text{satlimainvads1x}[[ix]] = \left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx} \right) * \left(\frac{\text{e1hostttandemads1x}[[ix]]}{\text{e1hostttandemTOTALx}[[ix]]} \right) * \text{sat1x}[[ix]], \{ix, nx\}];$$

```
Print[satlimainvads1x];
```

(* Dedido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo de Tx y una antena satelital. Entonces, Si existe por lo menos una central que emplee Tx satelital se hace lo siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima, y además una fracción proporcional del costo de la antena. Además, para imputar el costo a ADSL, se multiplica la inversión adicional estimada por la fracción de asignación, la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de ADSL entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

```
Print["satlimainvitxx"];
```

```
satlimainvitxx = zerox;
```

```
Do[If[numerosat == 0 ∨ e1hostttandemTOTALx[[ix]] == 0, satlimainvitxx[[ix]] = 0,
```

$$\text{satlimainvitxx}[[ix]] = \left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx} \right) * \left(\frac{\text{e1hostttandemitxx}[[ix]]}{\text{e1hostttandemTOTALx}[[ix]]} \right) * \text{sat1x}[[ix]], \{ix, nx\}];$$

```
Print[satlimainvitxx];
```

(* Dedido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo de Tx y una antena satelital. Entonces, Si existe por lo menos una central que emplee Tx satelital se hace lo siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima, y además una fracción proporcional del costo de la antena. Además, para imputar el costo a Circuitos de Interconexión, se multiplica la inversión adicional estimada por la fracción de asignación, la cual es equivalente al ratio entre el total de E1s satelitales a nivel nacional de Circuitos de Interconexión entre el total de E1s satelitales a nivel nacional que incluye todos los servicios *)

```
Print["satlimainvTOTALx"];
```

```
satlimainvTOTALx = zerox;
```

```
Do[If[numerosat == 0 ∨ e1hostttandemTOTALx[[ix]] == 0,
```

$$\text{satlimainvTOTALx}[[ix]] = 0, \text{satlimainvTOTALx}[[ix]] = \left(\frac{\text{satantena}}{\text{numerosat}} + \text{sateqtx} \right) * \text{sat1x}[[ix]], \{ix, nx\}];$$

```
Print[satlimainvitxx];
```

(* Dedido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo de Tx y una antena satelital. Entonces, Si existe por lo menos una central que emplee Tx satelital se hace lo siguiente: A cada nodo que usa Tx satelital se le imputa el respectivo equipo de Tx que esta en Lima, y además una fracción proporcional del costo de la antena. *)


```

Print["satlimainvd"];
satlimainvd = zerod;
Do[satlimainvd[[deptx[[ix]]] = satlimainvd[[deptx[[ix]]] + satlimainvx[[ix]], {ix, nx}]];
Print[satlimainvd];
(* Acumula para cada departamento la variable satlimainvx
correspondiente al plus de inversion adicional imputada a cada central que emplea
Tx satelital por los equipos que se requieren en Lima para el servicio de Voz *)

Print["satlimainvLocd"];
satlimainvLocd = zerod;
Do[satlimainvLocd[[deptx[[ix]]] = satlimainvLocd[[deptx[[ix]]] + satlimainvLocx[[ix]], {ix, nx}]];
Print[satlimainvLocd];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio Local de Voz *)

Print["satlimainvLDNd"];
satlimainvLDNd = zerod;
Do[satlimainvLDNd[[deptx[[ix]]] = satlimainvLDNd[[deptx[[ix]]] + satlimainvLDNx[[ix]], {ix, nx}]];
Print[satlimainvLDNd];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio LDN de Voz *)

Print["satlimainvLDId"];
satlimainvLDId = zerod;
Do[satlimainvLDId[[deptx[[ix]]] = satlimainvLDId[[deptx[[ix]]] + satlimainvLDIx[[ix]], {ix, nx}]];
Print[satlimainvLDId];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio LDI de Voz *)

Print["satlimainvalqdA1"];
satlimainvalqdA1 = zerod;
Do[satlimainvalqdA1[[deptx[[ix]]] = satlimainvalqdA1[[deptx[[ix]]] + satlimainvalqxA1[[ix]], {ix, nx}]];
Print[satlimainvalqdA1];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio de Circuitos Rango A *)

Print["satlimainvalqdB1"];
satlimainvalqdB1 = zerod;
Do[satlimainvalqdB1[[deptx[[ix]]] = satlimainvalqdB1[[deptx[[ix]]] + satlimainvalqxB1[[ix]], {ix, nx}]];
Print[satlimainvalqdB1];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio de Circuitos Rango B *)

Print["satlimainvalqdC1"];
satlimainvalqdC1 = zerod;
Do[satlimainvalqdC1[[deptx[[ix]]] = satlimainvalqdC1[[deptx[[ix]]] + satlimainvalqxC1[[ix]], {ix, nx}]];
Print[satlimainvalqdC1];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital
por los equipos que se requieren en Lima para el servicio de Circuitos Rango C *)

Print["satlimainvalqdLL"];
satlimainvalqdLL = zerod;
Do[satlimainvalqdLL[[deptx[[ix]]] = satlimainvalqdLL[[deptx[[ix]]] + satlimainvalqxLL[[ix]], {ix, nx}]];
Print[satlimainvalqdLL];
(* Acumula para cada departamento la variable satlimainvx correspondiente
al plus de inversion adicional imputada a cada central que emplea Tx satelital por
los equipos que se requieren en Lima para el servicio de Circuitos Local Lima *)

Print["satlimainvalqdLP"];
satlimainvalqdLP = zerod;
Do[satlimainvalqdLP[[deptx[[ix]]] = satlimainvalqdLP[[deptx[[ix]]] + satlimainvalqxLP[[ix]], {ix, nx}]];
Print[satlimainvalqdLP];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos Local Provincia *)

Print["satlimainvalqdAS"];
satlimainvalqdAS = zerod;
Do[satlimainvalqdAS[[deptx[[ix]]] = satlimainvalqdAS[[deptx[[ix]]] + satlimainvalqxAS[[ix]], {ix, nx}]];
Print[satlimainvalqdAS];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los

```

equipos que se requieren en Lima para el servicio de Circuitos Rango A Satelital*)

```
Print["satlimainvalqdBS"];
satlimainvalqdBS = zerod;
Do[satlimainvalqdBS[[deptx[[ix]]]] = satlimainvalqdBS[[deptx[[ix]]]] + satlimainvalqxBS[[ix]], {ix, nx}];
Print[satlimainvalqdBS];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos Rango B Satelital*)

Print["satlimainvalqdCS"];
satlimainvalqdCS = zerod;
Do[satlimainvalqdCS[[deptx[[ix]]]] = satlimainvalqdCS[[deptx[[ix]]]] + satlimainvalqxCS[[ix]], {ix, nx}];
Print[satlimainvalqdCS];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos Rango C Satelital*)

Print["satlimainvalqdABC2"];
satlimainvalqdABC2 = zerod;
Do[satlimainvalqdABC2[[deptx[[ix]]]] = satlimainvalqdABC2[[deptx[[ix]]]] + satlimainvalqxABC2[[ix]], {ix, nx}];
Print[satlimainvalqdABC2];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Otros Circuitos Rango A, B y C*)

Print["satlimainvalqdABCnop"];
satlimainvalqdABCnop = zerod;
Do[satlimainvalqdABCnop[[deptx[[ix]]]] = satlimainvalqdABCnop[[deptx[[ix]]]] + satlimainvalqxABCnop[[ix]], {ix, nx}];
Print[satlimainvalqdABCnop];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos de No Operadores*)

Print["satlimainvadsld"];
satlimainvadsld = zerod;
Do[satlimainvadsld[[deptx[[ix]]]] = satlimainvadsld[[deptx[[ix]]]] + satlimainvadsldx[[ix]], {ix, nx}];
Print[satlimainvadsld];
(* Acumula para cada departamento la variable satlimainvx
correspondiente al plus de inversion adicional imputada a cada central que emplea
Tx satelital por los equipos que se requieren en Lima para el servicio de ADSL *)

Print["satlimainvitxd"];
satlimainvitxd = zerod;
Do[satlimainvitxd[[deptx[[ix]]]] = satlimainvitxd[[deptx[[ix]]]] + satlimainvitxx[[ix]], {ix, nx}];
Print[satlimainvitxd];
(* Acumula para cada departamento la variable satlimainvx correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos de Interconexión*)

Print["satlimainvTOTALd"];
satlimainvTOTALd = zerod;
Do[satlimainvTOTALd[[deptx[[ix]]]] = satlimainvTOTALd[[deptx[[ix]]]] + satlimainvTOTALx[[ix]], {ix, nx}];
Print[satlimainvTOTALd];
(* Acumula para cada departamento la variable satlimainvx
correspondiente al plus de inversion adicional imputada a cada central que emplea
Tx satelital por los equipos que se requieren en Lima para todos los servicios*)

Print["satlimainv"];
satlimainv = Plus@@satlimainvd;
Print[satlimainv];
(* Acumula a nivel nacional la variable satlimainvd correspondiente al plus de inversion adicional imputada a
cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Voz *)

Print["satlimainvLoc"];
satlimainvLoc = Plus@@satlimainvLocd;
Print[satlimainvLoc];
(* Acumula a nivel nacional la variable satlimainvd correspondiente al plus de inversion adicional imputada a
cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio Local de Voz *)

Print["satlimainvLDN"];
satlimainvLDN = Plus@@satlimainvLDNd;
Print[satlimainvLDN];
(* Acumula a nivel nacional la variable satlimainvd correspondiente al plus de inversion adicional imputada a
cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio LDN de Voz *)

Print["satlimainvLDI"];
satlimainvLDI = Plus@@satlimainvLDId;
Print[satlimainvLDI];
```



```
(* Acumula a nivel nacional la variable satlimainvd correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio LDI de Voz *)
```

```
Print["satlimainvalqA1"];
```

```
satlimainvalqA1 = Plus@@satlimainvalqdA1;
```

```
Print[satlimainvalqA1];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdA1 correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango A *)
```

```
Print["satlimainvalqB1"];
```

```
satlimainvalqB1 = Plus@@satlimainvalqdB1;
```

```
Print[satlimainvalqB1];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdB1 correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango B *)
```

```
Print["satlimainvalqC1"];
```

```
satlimainvalqC1 = Plus@@satlimainvalqdC1;
```

```
Print[satlimainvalqC1];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdC1 correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango C *)
```

```
Print["satlimainvalqLL"];
```

```
satlimainvalqLL = Plus@@satlimainvalqdLL;
```

```
Print[satlimainvalqLL];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdLL correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Local Lima *)
```

```
Print["satlimainvalqLP"];
```

```
satlimainvalqLP = Plus@@satlimainvalqdLP;
```

```
Print[satlimainvalqLP];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdLP correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Local Provincias *)
```

```
Print["satlimainvalqAS"];
```

```
satlimainvalqAS = Plus@@satlimainvalqdAS;
```

```
Print[satlimainvalqAS];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdAS correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango A Satelital*)
```

```
Print["satlimainvalqBS"];
```

```
satlimainvalqBS = Plus@@satlimainvalqdBS;
```

```
Print[satlimainvalqBS];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdB1 correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango B Satelital*)
```

```
Print["satlimainvalqCS"];
```

```
satlimainvalqCS = Plus@@satlimainvalqdCS;
```

```
Print[satlimainvalqCS];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdCS correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos Rango C Satelital*)
```

```
Print["satlimainvalqABC2"];
```

```
satlimainvalqABC2 = Plus@@satlimainvalqdABC2;
```

```
Print[satlimainvalqABC2];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdA1nop correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Otros Circuitos Rango A, B y C *)
```

```
Print["satlimainvalqABCnop"];
```

```
satlimainvalqABCnop = Plus@@satlimainvalqdABCnop;
```

```
Print[satlimainvalqABCnop];
```

```
(* Acumula a nivel nacional la variable satlimainvalqdB1nop correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de Circuitos de No Operadores *)
```

```
Print["satlimainvads1"];
```

```
satlimainvads1 = Plus@@satlimainvads1d;
```

```
Print[satlimainvads1];
```

```
(* Acumula a nivel nacional la variable satlimainvads1d correspondiente al plus de inversion adicional imputada a cada central que emplea Tx satelital por los equipos que se requieren en Lima para el servicio de ADSL *)
```

```

Print["satlimainvitx"];
satlimainvitx = Plus @@ satlimainvitxd;
Print[satlimainvitx];
(* Acumula a nivel nacional la variable satlimainvitxd correspondiente al
plus de inversion adicional imputada a cada central que emplea Tx satelital por los
equipos que se requieren en Lima para el servicio de Circuitos de Interconexión *)

Print["satlimainvTOTAL"];
satlimainvTOTAL = Plus @@ satlimainvTOTALd;
Print[satlimainvTOTAL];
(* Acumula a nivel nacional la variable satlimainvTOTALd correspondiente al plus de inversion adicional imputada
a cada central que emplea Tx satelital por los equipos que se requieren en Lima para todos los servicios *)

Print[Style["A NIVEL DE ANCHO DE BANDA SATELITAL", 16, White, Bold, Background -> Blue]];
(* Els del servicio de Voz ya existe y está calculado previamente *)
Print["e1satalqxA1"];
e1satalqxA1 = zerox;
Do[e1satalqxA1[[inx]] = e1hostttandemalqxA1[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxA1];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma
la variable e1hostttandemalqxA1 equivalente a la carga en Els del servicio Circuitos Rango A *)

Print["e1satalqxB1"];
e1satalqxB1 = zerox;
Do[e1satalqxB1[[inx]] = e1hostttandemalqxB1[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxB1];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma
la variable e1hostttandemalqxB1 equivalente a la carga en Els del servicio Circuitos Rango B *)

Print["e1satalqxC1"];
e1satalqxC1 = zerox;
Do[e1satalqxC1[[inx]] = e1hostttandemalqxC1[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxC1];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma
la variable e1hostttandemalqxC1 equivalente a la carga en Els del servicio Circuitos Rango C *)

Print["e1satalqxLL"];
e1satalqxLL = zerox;
Do[e1satalqxLL[[inx]] = e1hostttandemalqxLL[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxLL];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hostttandemalqxLL equivalente a la carga en Els del servicio Circuitos Local Lima*)

Print["e1satalqxLP"];
e1satalqxLP = zerox;
Do[e1satalqxLP[[inx]] = e1hostttandemalqxLP[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxLP];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hostttandemalqxLP equivalente a la carga en Els del servicio Circuitos Local Provincia*)

Print["e1satalqxAS"];
e1satalqxAS = zerox;
Do[e1satalqxAS[[inx]] = e1hostttandemalqxAS[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxAS];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hostttandemalqxAS equivalente a la carga en Els del servicio Circuitos Rango A Satelital*)

Print["e1satalqxBs"];
e1satalqxBs = zerox;
Do[e1satalqxBs[[inx]] = e1hostttandemalqxBs[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxBs];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hostttandemalqxBs equivalente a la carga en Els del servicio Circuitos Rango B Satelital*)

Print["e1satalqxCS"];
e1satalqxCS = zerox;
Do[e1satalqxCS[[inx]] = e1hostttandemalqxCS[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxCS];
(* Identifica la carga en Els correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hostttandemalqxCS equivalente a la carga en Els del servicio Circuitos Rango C Satelital*)

```

```

Print["e1satalqxABC2"];
e1satalqxABC2 = zerox;
Do[e1satalqxABC2[[inx]] = e1hosttandemalqxABC2[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxABC2];
(* Identifica la carga en E1s correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la variable
e1hosttandemalqxABC2 equivalente a la carga en E1s del servicio Otros Circuitos Rango A, B y C*)

Print["e1satalqxABCnop"];
e1satalqxABCnop = zerox;
Do[e1satalqxABCnop[[inx]] = e1hosttandemalqxABCnop[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satalqxABCnop];
(* Identifica la carga en E1s correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la variable
e1hosttandemalqxABCnop equivalente a la carga en E1s del servicio Circuitos de No Operadores*)

Print["e1satadslx"];
e1satadslx = zerox;
Do[e1satadslx[[inx]] = e1hosttandemadslx[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satadslx];
(* Identifica la carga en E1s correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital
se toma la variable e1hosttandemadslx equivalente a la carga en E1s del servicio ADSL *)

Print["e1satitxx"];
e1satitxx = zerox;
Do[e1satitxx[[inx]] = e1hosttandemitxx[[inx]] * host1x[[inx]] * sat1x[[inx]] * (1 - tandem1x[[inx]]), {inx, nx}];
Print[e1satitxx];
(* Identifica la carga en E1s correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se toma la
variable e1hosttandemitxx equivalente a la carga en E1s del servicio Circuitos de Interconexión *)

Print["e1satTOTALx"];
e1satTOTALx = e1satx + e1satalqxA1 + e1satalqxB1 + e1satalqxC1 + e1satalqxLL + e1satalqxLP +
e1satalqxAS + e1satalqxBS + e1satalqxCS + e1satalqxABC2 + e1satalqxABCnop + e1satitxx + e1satadslx;
Print[e1satTOTALx];
(* Identifica la carga en E1s correspondientes a los nodos que emplean Tx satelital. Es decir,
sólo para las centrales Host que son distintas de la Tandem y que emplean Tx satelital se
toma la variable e1hosttandemTOTALx equivalente a la carga en E1s de todos los servicios *)

Print["Extrasatcostx"];
Extrasatcostx =  $\frac{2 \text{ Ceiling}[e1satx] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[Extrasatcostx];
(* Sobre la base del número de E1s satelitales se estima el gasto
directo en el alquiler del ancho de banda satelital. Servicio de Voz *)

Print["ExtrasatcostLocx"];
ExtrasatcostLocx = Extrasatcostx * satLocx;
Print[ExtrasatcostLocx];
(* Divide el Servicio de Voz en Local *)

Print["ExtrasatcostLDNx"];
ExtrasatcostLDNx = Extrasatcostx * satLDNx;
Print[ExtrasatcostLDNx];
(* Divide el Servicio de Voz en LDN *)

Print["ExtrasatcostLDIx"];
ExtrasatcostLDIx = Extrasatcostx * satLDIx;
Print[ExtrasatcostLDIx];
(* Divide el Servicio de Voz en LDI *)

Print["ExtrasatcostalqxA1"];
ExtrasatcostalqxA1 =  $\frac{2 \text{ Ceiling}[e1satalqxA1] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxA1];
(* Sobre la base del número de E1s satelitales se estima el gasto directo
en el alquiler del ancho de banda satelital. Servicio de Circuitos Rango A *)

Print["ExtrasatcostalqxB1"];
ExtrasatcostalqxB1 =  $\frac{2 \text{ Ceiling}[e1satalqxB1] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxB1];
(* Sobre la base del número de E1s satelitales se estima el gasto directo
en el alquiler del ancho de banda satelital. Servicio de Circuitos Rango B *)

Print["ExtrasatcostalqxC1"];
ExtrasatcostalqxC1 =  $\frac{2 \text{ Ceiling}[e1satalqxC1] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;

```

```

Print[ExtrasatcostalqxCl];
(* Sobre la base del número de E1s satelitales se estima el gasto directo
en el alquiler del ancho de banda satelital. Servicio de Circuitos Rango A *)

Print["ExtrasatcostalqxLL"];
ExtrasatcostalqxLL =  $\frac{2 \text{ Ceiling}[e1satalqxLL] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxLL];
(* Sobre la base del número de E1s satelitales se estima el gasto directo
en el alquiler del ancho de banda satelital. Servicio de Circuitos Local Lima *)

Print["ExtrasatcostalqxLP"];
ExtrasatcostalqxLP =  $\frac{2 \text{ Ceiling}[e1satalqxLP] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxLP];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos Local Provincia *)

Print["ExtrasatcostalqxAS"];
ExtrasatcostalqxAS =  $\frac{2 \text{ Ceiling}[e1satalqxAS] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxAS];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos Rango A Satelital*)

Print["ExtrasatcostalqxBS"];
ExtrasatcostalqxBS =  $\frac{2 \text{ Ceiling}[e1satalqxBS] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxBS];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos Rango B Satelital*)

Print["ExtrasatcostalqxCS"];
ExtrasatcostalqxCS =  $\frac{2 \text{ Ceiling}[e1satalqxCS] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxCS];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos Rango A Satelital*)

Print["ExtrasatcostalqxABC2"];
ExtrasatcostalqxABC2 =  $\frac{2 \text{ Ceiling}[e1satalqxABC2] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxABC2];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en el
alquiler del ancho de banda satelital. Servicio de Otros Circuitos Rango A, B y C*)

Print["ExtrasatcostalqxABCnop"];
ExtrasatcostalqxABCnop =  $\frac{2 \text{ Ceiling}[e1satalqxABCnop] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostalqxABCnop];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos de No Operadores*)

Print["Extrasatcostadslx"];
Extrasatcostadslx =  $\frac{2 \text{ Ceiling}[e1satadslx] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[Extrasatcostadslx];
(* Sobre la base del número de E1s satelitales se estima el gasto
directo en el alquiler del ancho de banda satelital. Servicio de ADSL *)

Print["Extrasatcostitxx"];
Extrasatcostitxx =  $\frac{2 \text{ Ceiling}[e1satitxx] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[Extrasatcostitxx];
(* Sobre la base del número de E1s satelitales se estima el gasto directo en
el alquiler del ancho de banda satelital. Servicio de Circuitos de Interconexión *)

Print["ExtrasatcostTOTALx"];
ExtrasatcostTOTALx =  $\frac{2 \text{ Ceiling}[e1satTOTALx] * \text{numMHzporMb} * \text{costesatporMHz}}{2}$ ;
Print[ExtrasatcostTOTALx];
(* Sobre la base del número de E1s satelitales se estima
el gasto directo en el alquiler del ancho de banda satelital. *)

Print["Extrasatcostd"];
Extrasatcostd = zerod;
Do[Extrasatcostd[[deptx[[ix]]]] = Extrasatcostd[[deptx[[ix]]]] + Extrasatcostx[[ix]], {ix, nx}];
Print[Extrasatcostd];
(* Acumula por departamento la variable Extrasatcostx, Servicio de Voz *) (*Ojo*)

```

```

Print["factor4"];
factor4 = zerox;

Do[ factor4[[ix]] = If[ linesd[[deptx[[ix]]] == linesx[[ix]], 0,  $\frac{\text{satlinesd}[[\text{deptx}[[ix]]] - \text{satlinesx}[[ix]]}{\text{linesd}[[\text{deptx}[[ix]]] - \text{linesx}[[ix]]}$  ];;, {ix, nx}];

Print[factor4];
(*Para igualar al propuesto, no se le quita el interno,
asumiendo que también viene hasta LIMA como en el Modelo de ACCESS*)

Print["satmind4"];
satmind4 = zerod;
Do[If[deptx[[ix]] ≠ MADREDEDIOS, satmind4[[deptx[[ix]]] =
    satmind4[[deptx[[ix]]] + (switchendx[[ix]](*-intrax[[ix]]*) - satldintradx[[ix]] * factor4[[ix]]) * sat1x[[ix]]];, {ix, nx}];
Print[satmind4];

Print["ExtrasatcostLocd"];
ExtrasatcostLocd = zerod;
Do[ExtrasatcostLocd[[deptx[[ix]]] = ExtrasatcostLocd[[deptx[[ix]]] + ExtrasatcostLocx[[ix]], {ix, nx}];
Print[ExtrasatcostLocd];
(* Acumula por departamento la variable ExtrasatcostLocx, Servicio Local de Voz *)

Print["ExtrasatcostLDNd"];
ExtrasatcostLDNd = zerod;
Do[ExtrasatcostLDNd[[deptx[[ix]]] = ExtrasatcostLDNd[[deptx[[ix]]] + ExtrasatcostLDNx[[ix]], {ix, nx}];
Print[ExtrasatcostLDNd];
(* Acumula por departamento la variable ExtrasatcostLDNx, Servicio LDN de Voz *)

Print["ExtrasatcostLDId"];
ExtrasatcostLDId = zerod;
Do[ExtrasatcostLDId[[deptx[[ix]]] = ExtrasatcostLDId[[deptx[[ix]]] + ExtrasatcostLDIx[[ix]], {ix, nx}];
Print[ExtrasatcostLDId];
(* Acumula por departamento la variable ExtrasatcostLDIx, Servicio LDI de Voz *)

Print["ExtrasatcostalqdA1"];
ExtrasatcostalqdA1 = zerod;
Do[ExtrasatcostalqdA1[[deptx[[ix]]] = ExtrasatcostalqdA1[[deptx[[ix]]] + ExtrasatcostalqxA1[[ix]], {ix, nx}];
Print[ExtrasatcostalqdA1];
(* Acumula por departamento la variable ExtrasatcostalqxA1, Servicio de Circuitos Rango A *)

Print["ExtrasatcostalqdB1"];
ExtrasatcostalqdB1 = zerod;
Do[ExtrasatcostalqdB1[[deptx[[ix]]] = ExtrasatcostalqdB1[[deptx[[ix]]] + ExtrasatcostalqxB1[[ix]], {ix, nx}];
Print[ExtrasatcostalqdB1];
(* Acumula por departamento la variable ExtrasatcostalqxB1, Servicio de Circuitos Rango B *)

Print["ExtrasatcostalqdC1"];
ExtrasatcostalqdC1 = zerod;
Do[ExtrasatcostalqdC1[[deptx[[ix]]] = ExtrasatcostalqdC1[[deptx[[ix]]] + ExtrasatcostalqxC1[[ix]], {ix, nx}];
Print[ExtrasatcostalqdC1];
(* Acumula por departamento la variable ExtrasatcostalqxC1, Servicio de Circuitos Rango C *)

Print["ExtrasatcostalqdLL"];
ExtrasatcostalqdLL = zerod;
Do[ExtrasatcostalqdLL[[deptx[[ix]]] = ExtrasatcostalqdLL[[deptx[[ix]]] + ExtrasatcostalqxLL[[ix]], {ix, nx}];
Print[ExtrasatcostalqdLL];
(* Acumula por departamento la variable ExtrasatcostalqxLL, Servicio de Circuitos Local Lima *)

Print["ExtrasatcostalqdLP"];
ExtrasatcostalqdLP = zerod;
Do[ExtrasatcostalqdLP[[deptx[[ix]]] = ExtrasatcostalqdLP[[deptx[[ix]]] + ExtrasatcostalqxLP[[ix]], {ix, nx}];
Print[ExtrasatcostalqdLP];
(* Acumula por departamento la variable ExtrasatcostalqxLP, Servicio de Circuitos Local Provincia *)

Print["ExtrasatcostalqdAS"];
ExtrasatcostalqdAS = zerod;
Do[ExtrasatcostalqdAS[[deptx[[ix]]] = ExtrasatcostalqdAS[[deptx[[ix]]] + ExtrasatcostalqxAS[[ix]], {ix, nx}];
Print[ExtrasatcostalqdAS];
(* Acumula por departamento la variable ExtrasatcostalqxAS, Servicio de Circuitos Rango A Satelital*)

Print["ExtrasatcostalqdBS"];
ExtrasatcostalqdBS = zerod;
Do[ExtrasatcostalqdBS[[deptx[[ix]]] = ExtrasatcostalqdBS[[deptx[[ix]]] + ExtrasatcostalqxBS[[ix]], {ix, nx}];
Print[ExtrasatcostalqdBS];
(* Acumula por departamento la variable ExtrasatcostalqxBS, Servicio de Circuitos Rango B Satelital*)

Print["ExtrasatcostalqdCS"];
ExtrasatcostalqdCS = zerod;
Do[ExtrasatcostalqdCS[[deptx[[ix]]] = ExtrasatcostalqdCS[[deptx[[ix]]] + ExtrasatcostalqxCS[[ix]], {ix, nx}];
Print[ExtrasatcostalqdCS];

```



```
(* Acumula por departamento la variable ExtrasatcostalqxCS, Servicio de Circuitos Rango C Satelital*)

Print["ExtrasatcostalqdABC2"];
ExtrasatcostalqdABC2 = zerod;
Do[ExtrasatcostalqdABC2[[deptx[ix]]] = ExtrasatcostalqdABC2[[deptx[ix]]] + ExtrasatcostalqxABC2[[ix]], {ix, nx}];
Print[ExtrasatcostalqdABC2];
(* Acumula por departamento la variable ExtrasatcostalqxABC2, Servicio de Otros Circuitos Rango A, B y C*)

Print["ExtrasatcostalqdABCnop"];
ExtrasatcostalqdABCnop = zerod;
Do[ExtrasatcostalqdABCnop[[deptx[ix]]] = ExtrasatcostalqdABCnop[[deptx[ix]]] + ExtrasatcostalqxABCnop[[ix]], {ix, nx}];
Print[ExtrasatcostalqdABCnop];
(* Acumula por departamento la variable ExtrasatcostalqxABCnop, Servicio de Circuitos de No Operadores*)

Print["Extrasatcostadsld"];
Extrasatcostadsld = zerod;
Do[Extrasatcostadsld[[deptx[ix]]] = Extrasatcostadsld[[deptx[ix]]] + Extrasatcostadslx[[ix]], {ix, nx}];
Print[Extrasatcostadsld];
(* Acumula por departamento la variable Extrasatcostadslx, Servicio de ADSL *)

Print["Extrasatcostitxd"];
Extrasatcostitxd = zerod;
Do[Extrasatcostitxd[[deptx[ix]]] = Extrasatcostitxd[[deptx[ix]]] + Extrasatcostitxx[[ix]], {ix, nx}];
Print[Extrasatcostitxd];
(* Acumula por departamento la variable Extrasatcostitxx, Servicio de Circuitos de Interconexión*)

Print["ExtrasatcostTOTALd"];
ExtrasatcostTOTALd = zerod;
Do[ExtrasatcostTOTALd[[deptx[ix]]] = ExtrasatcostTOTALd[[deptx[ix]]] + ExtrasatcostTOTALx[[ix]], {ix, nx}];
Print[ExtrasatcostTOTALd];
(* Acumula por departamento la variable ExtrasatcostTOTALx, Todos los servicios*)
```

ESTIMACIÓN DE COMPOSICION DE E1s

```
Print[Style["ESTIMACIÓN DE COMPOSICION DE E1s (si es del Anillo o
de tramo independiente)", 16, White, Bold, Underlined, RGBColor[0, 1/2, 0]]];

Print[Style["A NIVEL HOST TANDEM", 16, White, Bold, Underlined, Background->Blue]];

Print["e1compHTx"];
e1compHTx = zerox;
Do[If[esAnillodx[ix] ≠ NA, e1compHTx[ix] = e1Anillod[[deptx[ix]]], e1compHTx[ix] = e1hosttandemx[[ix]]];, {ix, nx}];
Print[e1compHTx];
(* Especifica el número de E1s relevante para el servicio de
Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental
se tomar la variable e1Anillod y si no es anillo simplemente la variable e1hosttandemx,
ambas referidas de manera exclusiva al servicio elegido *)

Print["loadcompHTLocx"];
loadcompHTLocx = zerox;
Do[If[esAnillodx[ix] ≠ NA, loadcompHTLocx[ix] = loadAnilloLocd[[deptx[ix]]],
loadcompHTLocx[ix] = loadhosttandemLocx[[ix]]];, {ix, nx}];
Print[loadcompHTLocx];
(* Especifica la carga en Erlangs relevante para el servicio Local de
Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
la variable loadAnilloLocd y si no es anillo simplemente la variable loadhosttandemLocx,
ambas referidas de manera exclusiva al servicio elegido *)

Print["loadcompHTLDNx"];
loadcompHTLDNx = zerox;
Do[If[esAnillodx[ix] ≠ NA, loadcompHTLDNx[ix] = loadAnilloLDNd[[deptx[ix]]],
loadcompHTLDNx[ix] = loadhosttandemLDNx[[ix]]];, {ix, nx}];
Print[loadcompHTLDNx];
(* Especifica la carga en Erlangs relevante para el servicio LDN de
Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
la variable loadAnilloLDNd y si no es anillo simplemente la variable loadhosttandemLDNx,
ambas referidas de manera exclusiva al servicio elegido *)

Print["loadcompHTLDIx"];
loadcompHTLDIx = zerox;
Do[If[esAnillodx[ix] ≠ NA, loadcompHTLDIx[ix] = loadAnilloLDId[[deptx[ix]]],
loadcompHTLDIx[ix] = loadhosttandemLDIx[[ix]]];, {ix, nx}];
Print[loadcompHTLDIx];
(* Especifica la carga en Erlangs relevante para el servicio LDI de
Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
la variable loadAnilloLDId y si no es anillo simplemente la variable loadhosttandemLDIx,
ambas referidas de manera exclusiva al servicio elegido *)
```



```

Print["loadcompHTx"];
loadcompHTx = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, loadcompHTx[[ix]] = loadAnilloalqA1[[deptx[[ix]]], loadcompHTx[[ix]] = loadhosttandemx[[ix]]],,
  {ix, nx}];
Print[loadcompHTx];
(* Especifica la carga en Erlangs relevante para el servicio de
  Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
  tomar la variable loadAnilloalqA1 y si no es anillo simplemente la variable loadhosttandemx,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxA1"];
e1compalqHTxA1 = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxA1[[ix]] = e1AnilloalqA1[[deptx[[ix]]],
  e1compalqHTxA1[[ix]] = e1hosttandemalqxA1[[ix]]],, {ix, nx}];
Print[e1compalqHTxA1];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango
  A: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable e1AnilloalqA1 y si no es anillo simplemente la variable e1hosttandemalqxA1,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxB1"];
e1compalqHTxB1 = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxB1[[ix]] = e1AnilloalqB1[[deptx[[ix]]],
  e1compalqHTxB1[[ix]] = e1hosttandemalqxB1[[ix]]],, {ix, nx}];
Print[e1compalqHTxB1];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango
  B: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable e1AnilloalqB1 y si no es anillo simplemente la variable e1hosttandemalqxB1,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxC1"];
e1compalqHTxC1 = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxC1[[ix]] = e1AnilloalqC1[[deptx[[ix]]],
  e1compalqHTxC1[[ix]] = e1hosttandemalqxC1[[ix]]],, {ix, nx}];
Print[e1compalqHTxC1];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango
  C: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable e1AnilloalqC1 y si no es anillo simplemente la variable e1hosttandemalqxC1,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxLL"];
e1compalqHTxLL = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxLL[[ix]] = e1AnilloalqLL[[deptx[[ix]]],
  e1compalqHTxLL[[ix]] = e1hosttandemalqxLL[[ix]]],, {ix, nx}];
Print[e1compalqHTxLL];
(* Especifica el número de E1s relevante para el servicio de Circuitos Local
  Lima: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable e1AnilloalqLL y si no es anillo simplemente la variable e1hosttandemalqxLL,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxLP"];
e1compalqHTxLP = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxLP[[ix]] = e1AnilloalqLP[[deptx[[ix]]],
  e1compalqHTxLP[[ix]] = e1hosttandemalqxLP[[ix]]],, {ix, nx}];
Print[e1compalqHTxLP];
(* Especifica el número de E1s relevante para el servicio de Circuitos Local
  Provincias: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
  tomar la variable e1AnilloalqLP y si no es anillo simplemente la variable e1hosttandemalqxLP,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxAS"];
e1compalqHTxAS = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxAS[[ix]] = e1AnilloalqAS[[deptx[[ix]]],
  e1compalqHTxAS[[ix]] = e1hosttandemalqxAS[[ix]]],, {ix, nx}];
Print[e1compalqHTxAS];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango A
  Satelital: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
  tomar la variable e1AnilloalqAS y si no es anillo simplemente la variable e1hosttandemalqxAS,
  ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxBS"];
e1compalqHTxBS = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxBS[[ix]] = e1AnilloalqBS[[deptx[[ix]]],
  e1compalqHTxBS[[ix]] = e1hosttandemalqxBS[[ix]]],, {ix, nx}];
Print[e1compalqHTxBS];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango B
  Satelital: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
  tomar la variable e1AnilloalqBS y si no es anillo simplemente la variable e1hosttandemalqxBS,

```

```
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxCS"];
e1compalqHTxCS = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxCS[[ix]] = e1AnilloalqdCS[[deptx[[ix]]],
    e1compalqHTxCS[[ix]] = e1hosttandemalqxCS[[ix]]];, {ix, nx}];
Print[e1compalqHTxCS];
(* Especifica el número de E1s relevante para el servicio de Circuitos Rango C
Satelital: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
tomar la variable e1AnilloalqdCS y si no es anillo simplemente la variable e1hosttandemalqxCS,
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxABC2"];
e1compalqHTxABC2 = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxABC2[[ix]] = e1AnilloalqdABC2[[deptx[[ix]]],
    e1compalqHTxABC2[[ix]] = e1hosttandemalqxABC2[[ix]]];, {ix, nx}];
Print[e1compalqHTxABC2];
(* Especifica el número de E1s relevante para el servicio de Otros Circuitos Rango A,
B y C: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
la variable e1AnilloalqdABC2 y si no es anillo simplemente la variable e1hosttandemalqxABC2,
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compalqHTxABCnop"];
e1compalqHTxABCnop = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compalqHTxABCnop[[ix]] = e1AnilloalqdABCnop[[deptx[[ix]]],
    e1compalqHTxABCnop[[ix]] = e1hosttandemalqxABCnop[[ix]]];, {ix, nx}];
Print[e1compalqHTxABCnop];
(* Especifica el número de E1s relevante para el servicio de Circuitos de No
Operadores: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
la variable e1AnilloalqdABCnop y si no es anillo simplemente la variable e1hosttandemalqxABCnop,
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compitxHTx"];
e1compitxHTx = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compitxHTx[[ix]] = Round[e1Anilloitxd[[deptx[[ix]]]],
    e1compitxHTx[[ix]] = Round[e1hosttandemitxx[[ix]]];, {ix, nx}];
Print[e1compitxHTx];
(* Especifica el número de E1s relevante para el servicio de Circuitos de
Interconexión: Vale para radio y fibra. Si la central forma parte de un anillo departamental
se tomar la variable e1Anilloitxd y si no es anillo simplemente la variable e1hosttandemitxx,
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compadslHTx"];
e1compadslHTx = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compadslHTx[[ix]] = Round[e1Anilloadsld[[deptx[[ix]]]],
    e1compadslHTx[[ix]] = Round[e1hosttandemadslx[[ix]]];, {ix, nx}];
Print[e1compadslHTx];
(* Especifica el número de E1s relevante para el servicio de
ADSL: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
tomar la variable e1Anilloadsld y si no es anillo simplemente la variable e1hosttandemadslx,
ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compHTTOTALx"];
e1compHTTOTALx = zerox;
Do[If[esAnillodx[[ix]] ≠ NA, e1compHTTOTALx[[ix]] = e1AnilloTOTALd[[deptx[[ix]]],
    e1compHTTOTALx[[ix]] = e1hosttandemTOTALx[[ix]]];, {ix, nx}];
Print[e1compHTTOTALx];
(* Especifica el número de E1s totales (de todos los servicios):
Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar la
variable e1AnilloTOTALd y si no es anillo simplemente la variable e1hosttandemTOTALx *)

Print[Style["COMPOSICIÓN POR SERVICIO = HOST TANDEM",
    16, White, Bold, Underlined, Background → Blue]];

Pcte1compHTx = Table[0, {jx, nx}, {ix, 13}];
Do[If[e1compHTTOTALx[[ix]] ≠ 0,
    Pcte1compHTx[[ix, 1]] =  $\frac{e1compHTx[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
    Pcte1compHTx[[ix, 2]] =  $\frac{e1compalqHTxA1[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
    Pcte1compHTx[[ix, 3]] =  $\frac{e1compalqHTxB1[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
    Pcte1compHTx[[ix, 4]] =  $\frac{e1compalqHTxC1[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
    Pcte1compHTx[[ix, 5]] =  $\frac{e1compalqHTxLL[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
```

```
Pcte1compHTx[[ix, 6]] =  $\frac{e1compalqHTxLP[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 7]] =  $\frac{e1compalqHTxAS[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 8]] =  $\frac{e1compalqHTxBS[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 9]] =  $\frac{e1compalqHTxCS[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 10]] =  $\frac{e1compalqHTxABC2[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 11]] =  $\frac{e1compalqHTxABCnop[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 12]] =  $\frac{e1compitxHTx[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
Pcte1compHTx[[ix, 13]] =  $\frac{e1compadslHTx[[ix]]}{e1compHTTOTALx[[ix]]}$ ; ],, {ix, nx}];
Print[TableForm[Pcte1compHTx, TableAlignments → {Center, Center}, TableDirections → {Column, Row},
  TableHeadings → {Table[ix, {ix, nx}], {"Voz", "Alq A", "Alq B", "Alq C", "Loc Lima",
    "Loc Prov", "Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq Otros", "No Op", "Itx", "ADSL"}}]];

```

```
Print[Style["ESTIMACIÓN DE COMPOSICION DE E1s",
  16, Bold, White, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
Print[Style["A NIVEL REMOTE HOST", 16, Bold, White, Underlined, Background → Blue]];

Print["Cargas de Voz"];
loadcompRHx = zerox;
loadcompRHLocx = zerox;
loadcompRHLDNx = zerox;
loadcompRHLDIx = zerox;
Do[If[esAnillopx[[ix]] == 1, loadcompRHx[[ix]] = loadAnillop[[provx[[ix]]];
  loadcompRHLocx[[ix]] = loadAnilloLocp[[provx[[ix]]];
  loadcompRHLDNx[[ix]] = loadAnilloLDNp[[provx[[ix]]];
  loadcompRHLDIx[[ix]] = loadAnilloLDIp[[provx[[ix]]];, loadcompRHx[[ix]] = loadremotehostx[[ix]];
  loadcompRHLocx[[ix]] = loadremotehostLocx[[ix]];
  loadcompRHLDNx[[ix]] = loadremotehostLDNx[[ix]];
  loadcompRHLDIx[[ix]] = loadremotehostLDIx[[ix]]];],, {ix, nx}];
(* Especifica la carga en Erlangs relevante para el servicio de
  Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se
  tomar la variable loadAnillop y si no es anillo simplemente la variable loadremotehostx,
  ambas referidas de manera exclusiva al servicio elegido *)
(* Especifica la carga en Erlangs relevante para el servicio Local de
  Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable loadAnilloLocp y si no es anillo simplemente la variable loadremotehostLocx,
  ambas referidas de manera exclusiva al servicio elegido *)
(* Especifica la carga en Erlangs relevante para el servicio LDN de
  Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable loadAnilloLDNp y si no es anillo simplemente la variable loadremotehostLDNx,
  ambas referidas de manera exclusiva al servicio elegido *)
(* Especifica la carga en Erlangs relevante para el servicio LDI de
  Voz: Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar
  la variable loadAnilloLDIp y si no es anillo simplemente la variable loadremotehostLDIx,
  ambas referidas de manera exclusiva al servicio elegido *)
Print[TableForm[{loadcompRHx, loadcompRHLocx, loadcompRHLDNx, loadcompRHLDIx},
  TableAlignments → {Center, Center}, TableDirections → {Row, Column}, TableSpacing → {0, 0},
  TableHeadings → {"Total", "Local", "LDN", "LDI"}, Table[ix, {ix, nx}]]];
Print["E1s por Servicio"];
e1compRHx = zerox;
e1compalqRHxA1 = zerox;
e1compalqRHxB1 = zerox;
e1compalqRHxC1 = zerox;
e1compalqRHxLL = zerox;
e1compalqRHxLP = zerox;
e1compalqRHxAS = zerox;
e1compalqRHxBs = zerox;
e1compalqRHxCS = zerox;
e1compalqRHxABC2 = zerox;
e1compalqRHxABCnop = zerox;
e1compitxRHx = zerox;
e1compadslRHx = zerox;
e1compRHTOTALx = zerox;
Do[If[esAnillopx[[ix]] == 1, e1compRHx[[ix]] = e1Anillop[[provx[[ix]]];
  e1compalqRHxA1[[ix]] = e1AnilloalqpA1[[provx[[ix]]];
  e1compalqRHxB1[[ix]] = e1AnilloalqpB1[[provx[[ix]]];
  e1compalqRHxC1[[ix]] = e1AnilloalqpC1[[provx[[ix]]];
  e1compalqRHxLL[[ix]] = e1AnilloalqpLL[[provx[[ix]]];

```

```

e1compalqRHxLP[[ix]] = e1AnilloalqpLP[[provx[[ix]]];
e1compalqRHxAS[[ix]] = e1AnilloalqpAS[[provx[[ix]]];
e1compalqRHxBS[[ix]] = e1AnilloalqpBS[[provx[[ix]]];
e1compalqRHxCS[[ix]] = e1AnilloalqpCS[[provx[[ix]]];
e1compalqRHxABC2[[ix]] = e1AnilloalqpABC2[[provx[[ix]]];
e1compalqRHxABCnop[[ix]] = e1AnilloalqpABCnop[[provx[[ix]]];
e1compitxRHx[[ix]] = e1Anilloitxp[[provx[[ix]]];
e1compadslRHx[[ix]] = e1Anilloadslp[[provx[[ix]]];
e1compRHTOTALx[[ix]] = e1AnilloTOTALp[[provx[[ix]]];, e1compRHx[[ix]] = e1remotehostx[[ix]];
e1compalqRHxA1[[ix]] = e1remotehostalqx A1[[ix]];
e1compalqRHxB1[[ix]] = e1remotehostalqx B1[[ix]];
e1compalqRHxC1[[ix]] = e1remotehostalqx C1[[ix]];
e1compalqRHxLL[[ix]] = e1remotehostalqx LL[[ix]];
e1compalqRHxLP[[ix]] = e1remotehostalqx LP[[ix]];
e1compalqRHxAS[[ix]] = e1remotehostalqx AS[[ix]];
e1compalqRHxBS[[ix]] = e1remotehostalqx BS[[ix]];
e1compalqRHxCS[[ix]] = e1remotehostalqx CS[[ix]];
e1compalqRHxABC2[[ix]] = e1remotehostalqx ABC2[[ix]];
e1compalqRHxABCnop[[ix]] = e1remotehostalqx ABCnop[[ix]];
e1compitxRHx[[ix]] = e1remotehostitxx[[ix]];
e1compadslRHx[[ix]] = e1remotehostadslx[[ix]];
e1compRHTOTALx[[ix]] = e1remotehostTOTALx[[ix]];];, {ix, nx}];

(* Especifica el número de E1s relevante para los servicios de Voz,
Circuitos Rango A, Circuitos Rango B, Circuitos Rango C, Otros Circuitos Rango A,
B y C, Circuitos de No Operadores, Circuitos de Interconexión, ADSL y Totales:
Vale para radio y fibra. Si la central forma parte de un anillo departamental se tomar las variables e1Anillop,
e1AnilloalqpA1, e1AnilloalqpB1, e1AnilloalqpC1, e1AnilloalqpLL, e1AnilloalqpLP, e1AnilloalqpAS,
e1AnilloalqpBS, e1AnilloalqpCS, e1AnilloalqpABC2, e1AnilloalqpABCnop, e1Anilloitxp,
e1Anilloadslp o e1AnilloTOTALp y si no es anillo simplemente la variable e1remotehostx, e1remotehostalqx A1,
e1remotehostalqx B1, e1remotehostalqx C1, e1remotehostalqx LL, e1remotehostalqx LP, e1remotehostalqx AS,
e1remotehostalqx BS, e1remotehostalqx CS, e1remotehostalqx ABC2, e1remotehostalqx ABCnop, e1remotehostitxx,
e1remotehostadslx o e1remotehostTOTALx, ambas referidas de manera exclusiva al servicio respectivo *)
Print[TableForm[{e1compRHx, e1compalqRHxA1, e1compalqRHxB1, e1compalqRHxC1,
e1compalqRHxABC2, e1compalqRHxABCnop, e1compitxRHx, e1compadslRHx, e1compRHTOTALx},
TableAlignments -> {Center, Center}, TableDirections -> {Row, Column}, TableSpacing -> {0, 0},
TableHeadings -> {{ "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov", "Alq A Sat",
"Alq B Sat", "Alq C Sat", "Alq Otros", "No Op", "Itx", "ADSL", "Total"}, Table[ix, {ix, nx}]]]];

Print[Style["COMPOSICIÓN POR SERVICIO = REMOTE HOST",
16, Bold, White, Underlined, Background -> Blue]];

Pcte1compRHx = Table[0, {jx, nx}, {ix, 13}];
Do[If[e1compRHTOTALx[[ix]] != 0,
Pcte1compRHx[[ix, 1]] =  $\frac{e1compRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 2]] =  $\frac{e1compalqRHxA1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 3]] =  $\frac{e1compalqRHxB1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 4]] =  $\frac{e1compalqRHxC1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 5]] =  $\frac{e1compalqRHxLL[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 6]] =  $\frac{e1compalqRHxLP[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 7]] =  $\frac{e1compalqRHxAS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 8]] =  $\frac{e1compalqRHxBS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 9]] =  $\frac{e1compalqRHxCS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 10]] =  $\frac{e1compalqRHxABC2[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 11]] =  $\frac{e1compalqRHxABCnop[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 12]] =  $\frac{e1compitxRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;
Pcte1compRHx[[ix, 13]] =  $\frac{e1compadslRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ];];, {ix, nx}];

Print[
TableForm[Pcte1compRHx, TableAlignments -> {Center, Center}, TableDirections -> {Column, Row}, TableSpacing -> {0, 0},
TableHeadings -> {Table[ix, {ix, nx}], {"Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov",
"Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq Otros", "No Op", "Itx", "ADSL", "Total"}}]];

```

```
Print[Style["ESTIMACIÓN DE INVERSIONES EN RADIO A NIVEL CABECERA-TANDEM",
```

```
16, Bold, White, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
```

```
hophosttandemx = Floor[ $\frac{\text{disttandemx}}{\text{hop}}$ ];
```

```
(* Estima el número de Repetidoras: Equivale a la distancia relevante para cada arista entre la distancia máxima que debe existir entre cada repetidora. Para llegar al número de repetidoras es necesario restar 1 de dicho resultado *)
```

```
numtributariosRHTx = Ceiling[ $\frac{\text{e1hosttandemTOTALx}}{21}$ ];
```

```
(* Identifica el número de tributarios por central *)
```

```
VelocidadRHTx = zerox;
```

```
CosteFijoRHTx = zerox;
```

```
CosteTributariosRHTx = zerox;
```

```
CosteRepetidoresRHTx = zerox;
```

```
Do[If[e1compHTTtotalx[[ix]] ≠ 0,
```

```
Do[If[e1compHTTtotalx[[ix]] > TxRadioLimites[[i]], VelocidadRHTx[[ix]] = i];, {i, Length[TxRadioLimites]}];
```

```
If[VelocidadRHTx[[ix]] < Length[TxRadioLimites], VelocidadRHTx[[ix]] = VelocidadRHTx[[ix]] + 1];
```

```
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)
```

```
If[VelocidadRHTx[[ix]] ≠ 0, CosteFijoRHTx[[ix]] = TxRadioCosteFijo[VelocidadRHTx[[ix]]];
```

```
(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)
```

```
CosteTributariosRHTx[[ix]] = TxRadioCosteTributario[VelocidadRHTx[[ix]]];
```

```
(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)
```

```
CosteRepetidoresRHTx[[ix]] = TxRadioCosteRep[VelocidadRHTx[[ix]] * hophosttandemx[[ix]];];
```

```
(* Especifica el costo del equipo repetidor asociado a la velocidad o capacidad requerida*)
```

```
, {ix, nx}];
```

```
radioinvhosttandemx = zerox;
```

```
radioinvhosttandemalqxA1 = zerox;
```

```
radioinvhosttandemalqxB1 = zerox;
```

```
radioinvhosttandemalqxC1 = zerox;
```

```
radioinvhosttandemalqxLL = zerox;
```

```
radioinvhosttandemalqxLP = zerox;
```

```
radioinvhosttandemalqxAS = zerox;
```

```
radioinvhosttandemalqxBs = zerox;
```

```
radioinvhosttandemalqxCS = zerox;
```

```
radioinvhosttandemalqxABC2 = zerox;
```

```
radioinvhosttandemalqxABCnop = zerox;
```

```
radioinvhosttandemadslx = zerox;
```

```
radioinvhosttandemitxx = zerox;
```

```
radioinvhosttandemTOTALx = zerox;
```

```
Do[If[e1compHTTtotalx[[ix]] ≠ 0 ∧ tandem1x[[ix]] ≠ 1, radioinvhosttandemTOTALx[[ix]] =
```

```
2 * CosteFijoRHTx[[ix]] + CosteRepetidoresRHTx[[ix]] + 2 * CosteTributariosRHTx[[ix]] * numtributariosRHTx[[ix]];
```

```
(* Estima el nivel de inversión Total e imputable a todos los servicios. Si la central es Tandem el resultado debe ser cero (En el nivel HT todos los costos estan imputados a las cabeceras *)
```

```
radioinvhosttandemx[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compHTx}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxA1[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxA1}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxB1[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxB1}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxC1[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxC1}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxLL[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxLL}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxLP[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxLP}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxAS[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxAS}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxBs[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxBs}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxCS[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxCS}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxABC2[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxABC2}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemalqxABCnop[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compalqHTxABCnop}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemadslx[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compadslHTx}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ;
```

```
radioinvhosttandemitxx[[ix]] = radioinvhosttandemTOTALx[[ix]] *  $\frac{\text{e1compitxHTx}[[ix]]}{\text{e1compHTTtotalx}[[ix]]}$ ];
```

```
, {ix, nx}];
```

```
(* Estima el nivel de inversión Total e imputable al servicio de Voz. Para ello, utiliza como factor o proporción imputable el ratio entre las variables
```



```

(Especifica el número de E1s relevante para el servicio de Voz) y la variable e1compHTTOTALx
(Especifica el número de E1s de todos los servicios). Si la central es Tandem el resultado
debe ser cero (En el nivel HT todos los costos estan imputados a las cabeceras *)
(* Estima el nivel de inversión Total e imputable a cada servicio. Para ello,
utiliza como factor o proporción imputable el ratio entre las variables e1compHTx,
e1compalqHTxA1, e1compalqHTxB1, e1compalqHTxC1,e1compalqHTxLL, e1compalqHTxLP,
e1compalqHTxAS, e1compalqHTxBS, e1compalqHTxCS, e1compalqHTxABC2, e1compalqHTxABCnop,
e1compadslHTx y e1compitxHTx (Especifica el número de E1s relevante para el servicio de Cada Servicio)
y la variable e1compHTTOTALx (Especifica el número de E1s de todos los servicios). Si la central es
Tandem el resultado debe ser cero (En el nivel HT todos los costos estan imputados a las cabeceras *)
Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], hophosttandemx[[ix]], numtributariosRHTx[[ix]],
  VelocidadRHTx[[ix]], CosteFijoRHTx[[ix]], CosteTributariosRHTx[[ix]], CosteRepetidoresRHTx[[ix]],
  radioinvhosttandemx[[ix]], radioinvhosttandemalqxA1[[ix]], radioinvhosttandemalqxB1[[ix]],
  radioinvhosttandemalqxC1[[ix]], radioinvhosttandemalqxLL[[ix]], radioinvhosttandemalqxLP[[ix]],
  radioinvhosttandemalqxAS[[ix]], radioinvhosttandemalqxBs[[ix]], radioinvhosttandemalqxCs[[ix]],
  radioinvhosttandemalqxABC2[[ix]], radioinvhosttandemalqxABCnop[[ix]], radioinvhosttandemadslx[[ix]],
  radioinvhosttandemitxx[[ix]], radioinvhosttandemTOTALx[[ix]]}, {ix, nx}],
TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
{None, {"ix", "idx", "Nombre Central", "hophosttandemx", "numtributariosRHTx", "VelocidadRHTx", "CosteFijoRHTx",
"CosteTributariosRHTx", "CosteRepetidoresRHTx", "radioinvhosttandemx", "radioinvhosttandemalqxA1",
"radioinvhosttandemalqxB1", "radioinvhosttandemalqxC1", "radioinvhosttandemalqxLL",
"radioinvhosttandemalqxLP", "radioinvhosttandemalqxAS", "radioinvhosttandemalqxBs",
"radioinvhosttandemalqxCs", "radioinvhosttandemalqxABC2", "radioinvhosttandemalqxABCnop",
"radioinvhosttandemadslx", "radioinvhosttandemitxx", "radioinvhosttandemTOTALx"}}]];

```

```

Print[Style["ESTIMACIÓN DE INVERSIONES EN RADIO A NIVEL REMOTA-CABECERA",
16, Bold, White, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

```

```
hopremotehostx = zerox;
```

```
Do[hopremotehostx[[ix]] = If[Ceiling[ $\frac{\text{disthostx}[[ix]]}{\text{hop}}$ ] > 1, Ceiling[ $\frac{\text{disthostx}[[ix]]}{\text{hop}}$ ] - 1, 0];, {ix, nx}];
```

(* Estima el número de Repetidoras: Equivale a la distancia relevante para cada arista entre la distancia máxima que debe existir entre cada repetidora. Para llegar al número de repetidoras es necesario restar 1 de dicho resultado *)

```
numtributariosRRHx = Ceiling[ $\frac{\text{e1remotehostTOTALx}}{21}$ ];
```

(* Identifica el número de tributarios por central *)

```
VelocidadRRHx = zerox;
```

```
CosteFijoRRHx = zerox;
```

```
CosteTributariosRRHx = zerox;
```

```
CosteRepetidoresRRHx = zerox;
```

```

Do[If[e1compRHTOTALx[[ix]] ≠ 0,
  Do[If[e1compRHTOTALx[[ix]] > TxRadioLimites[[i]], VelocidadRRHx[[ix]] = i];, {i, Length[TxRadioLimites]}];
  If[VelocidadRRHx[[ix]] < Length[TxRadioLimites], VelocidadRRHx[[ix]] = VelocidadRRHx[[ix]] + 1];];
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)
If[VelocidadRRHx[[ix]] ≠ 0, CosteFijoRRHx[[ix]] = TxRadioCosteFijo[VelocidadRRHx[[ix]]];
  (* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)
  CosteTributariosRRHx[[ix]] = TxRadioCosteTributario[VelocidadRRHx[[ix]]];
  (* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)
  CosteRepetidoresRRHx[[ix]] = TxRadioCosteRep[VelocidadRRHx[[ix]]] * hopremotehostx[[ix]];];
(* Especifica el costo del equipo repetidor asociado a la velocidad o capacidad requerida*)
, {ix, nx}];

```

```
radioinvremotehostx = zerox;
```

```
radioinvremotehostalqxA1 = zerox;
```

```
radioinvremotehostalqxB1 = zerox;
```

```
radioinvremotehostalqxC1 = zerox;
```

```
radioinvremotehostalqxLL = zerox;
```

```
radioinvremotehostalqxLP = zerox;
```

```
radioinvremotehostalqxAS = zerox;
```

```
radioinvremotehostalqxBs = zerox;
```

```
radioinvremotehostalqxCs = zerox;
```

```
radioinvremotehostalqxABC2 = zerox;
```

```
radioinvremotehostalqxABCnop = zerox;
```

```
radioinvremotehostadslx = zerox;
```

```
radioinvremotehostitxx = zerox;
```

```
radioinvremotehostTOTALx = zerox;
```

```

Do[
  If[e1compRHTOTALx[[ix]] ≠ 0 ∧ tandem1x[[ix]] ≠ 1,
    radioinvremotehostTOTALx[[ix]] =
      2 CosteFijoRRHx[[ix]] + CosteRepetidoresRRHx[[ix]] + 2 CosteTributariosRRHx[[ix]] * numtributariosRRHx[[ix]];
    (* Estima el nivel de inversión Total e imputable a todos los servicios. Si la central es Tandem
    el resultado debe ser cero (En el nivel RH todos los costos estan imputados a las remotas) *)

```



```
radioinvremotehostx[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxA1[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxA1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxB1[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxB1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxC1[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxC1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxLL[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxLL[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxLP[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxLP[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxAS[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxAS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxBs[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxBs[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxCS[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxCS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxABC2[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxABC2[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostalqxABCnop[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compalqRHxABCnop[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostadslx[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compadslRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

radioinvremotehostitxx[[ix]] = radioinvremotehostTOTALx[[ix]] *  $\frac{e1compitxRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ];

, {ix, nx}];

(* Estima el nivel de inversión Total e imputable a cada servicio. Para ello,
utiliza como factor o proporción imputable el ratio entre las variables e1compRHx,
e1compalqRHxA1, e1compalqRHxB1, e1compalqRHxC1, e1compalqRHxLL, e1compalqRHxLP,
e1compalqRHxAS, e1compalqRHxBs, e1compalqRHxCS, e1compalqRHxABC2, e1compalqRHxABCnop,
e1compadslRHx y e1compitxRHx (Especifica el número de EIs relevante para cada servicio) y la variable
e1compRHTOTALx (Especifica el número de EIs de todos los servicios). Si la central es Tandem
el resultado debe ser cero (En el nivel RH todos los costos estan imputados a las remotas) *)

Print[TableForm[
  Table[{ix, idx[[ix]], y[[ix, 13]], hopremotehostx[[ix]], numtributariosRRHx[[ix]], VelocidadRRHx[[ix]], CosteFijoRRHx[[ix]],
    CosteTributariosRRHx[[ix]], CosteRepetidoresRRHx[[ix]], radioinvremotehostx[[ix]], radioinvremotehostalqxA1[[ix]],
    radioinvremotehostalqxB1[[ix]], radioinvremotehostalqxC1[[ix]], radioinvremotehostalqxLL[[ix]],
    radioinvremotehostalqxLP[[ix]], radioinvremotehostalqxAS[[ix]], radioinvremotehostalqxBs[[ix]],
    radioinvremotehostalqxCS[[ix]], radioinvremotehostalqxABC2[[ix]], radioinvremotehostalqxABCnop[[ix]],
    radioinvremotehostadslx[[ix]], radioinvremotehostitxx[[ix]], radioinvremotehostTOTALx[[ix]]}, {ix, nx}],
  TableDirections -> {Column, Row}, TableSpacing -> {0, 1}, TableHeadings ->
  {None, {"ix", "idx", "Nombre Central", "hopremotehostx", "numtributariosRRHx", "VelocidadRRHx", "CosteFijoRRHx",
    "CosteTributariosRRHx", "CosteRepetidoresRRHx", "radioinvremotehostx", "radioinvremotehostalqxA1",
    "radioinvremotehostalqxB1", "radioinvremotehostalqxC1", "radioinvremotehostalqxLL",
    "radioinvremotehostalqxLP", "radioinvremotehostalqxAS", "radioinvremotehostalqxBs",
    "radioinvremotehostalqxCS", "radioinvremotehostalqxABC2", "radioinvremotehostalqxABCnop",
    "radioinvremotehostadslx", "radioinvremotehostitxx", "radioinvremotehostTOTALx"}}]]];
```

CALCULO DE INVERSIÓN EN FIBRA

```
Print[Style["CALCULO DEL COSTO POR KM DE FIBRA",
  16, Bold, White, Underlined, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["TxFibraCosteKmUrban: Fibra Canalizada"];
TxFibraCosteKmUrban =
  compfiburb *  $\left( \frac{TxFibraCosteEmpalme}{TxFibraDistEntreEmpalmesKm} + TxFibraCosteCanalizacionKm + \frac{TxFibraCosteCamara}{TxFibraDistEntreCamarasKm} \right)$ ;
Print[TxFibraCosteKmUrban];
Print["TxFibraCosteKmEnterrado: Fibra Enterrada"];
TxFibraCosteKmEnterrado =  $\frac{TxFibraCosteEmpalme}{TxFibraDistEntreEmpalmesKm}$  + TxFibraCosteTrituboKm;
Print[TxFibraCosteKmEnterrado];
Print["TxFibraCosteKmAereo: Fibra Aerea"];
TxFibraCosteKmAereo = 4451.85900092466; (*Se toma de los preciaris de ACCESS*)
Print[TxFibraCosteKmAereo];
```

```
Print[Style["ESTIMACIÓN DE INVERSIONES EN FIBRA A NIVEL CABECERA_TANDEM",
  16, Bold, White, Underlined, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

repfibra2x = Floor[ $\frac{repfibrax}{hopfibra}$ ] + Floor[ $\frac{SumaCabecerax}{60}$ ];
```

(*Se le agrega SumaCabecerax para evitar no incluir los repetidores de cabecera y evitar colocar inexistentes*)
 (* Especifica el número de repetidoras de Fibra por tramo *)

numtributariosFHTx = Ceiling $\left[\frac{e1hosttandemTOTALx}{21}\right]$;

(* Identifica el número de tributarios por central *)

VelocidadFHTx = zerox;

CostePorKmFHTx = zerox;

CosteFijoFHTx = zerox;

CosteTributariosFHTx = zerox;

Do[If[e1compHTTOTALx[[ix]] ≠ 0,
 Do[If[e1compHTTOTALx[[ix]] > TxFibraLimites[[i]], VelocidadFHTx[[ix]] = i];, {i, Length[TxFibraLimites]}];
 If[VelocidadFHTx[[ix]] < Length[TxFibraLimites], VelocidadFHTx[[ix]] = VelocidadFHTx[[ix]] + 1];];

(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)

If[transtechx[[ix]] == Urbana, CostePorKmFHTx[[ix]] = TxFibraCosteKmUrban;, If[transtechx[[ix]] == Enterrado,
 CostePorKmFHTx[[ix]] = TxFibraCosteKmEnterrado;, CostePorKmFHTx[[ix]] = TxFibraCosteKmAereo;];];

(* Especifica el costo asociado a la obra civil dependiendo del tipo de Fibra *)

If[VelocidadFHTx[[ix]] ≠ 0, CosteFijoFHTx[[ix]] = TxFibraCosteFijo[VelocidadFHTx[[ix]]];

(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)

CosteTributariosFHTx[[ix]] = TxFibraCosteTributario[VelocidadFHTx[[ix]]];

(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

, {ix, nx}];

CosteExtraFHTx = zerox;

CosteExtra2FHTx = zerox;

Do[If[esAnillodx[[AnilloCabeceras[[ip]]]] ≠ NA ∧ e1compHTTOTALx[[AnilloCabeceras[[ip]]]] ≠ 0,

CosteExtraFHTx[[AnilloCabeceras[[ip]]]] =

$\left(\text{CosteFijoFHTx}[[\text{AnilloCabeceras}[[ip]]]] + \text{CosteTributariosFHTx}[[\text{AnilloCabeceras}[[ip]]]] * \right.$
 $\left. \text{Ceiling}\left[\frac{e1compHTTOTALx[[AnilloCabeceras[[ip]]]]}{21}\right] \right) * \frac{e1hosttandemTOTALx[[AnilloCabeceras[[ip]]]]}{e1compHTTOTALx[[AnilloCabeceras[[ip]]]]};$

(* A nivel HT todos los costos deben ser imputados a las cabeceras no Tandem. En ese sentido, cuando hay anillo HT, los costos de los equipos de Tx y de los tributarios de cada una de las Tandem deben ser repartidos entre todas las centrales cabeceras que forman parte del anillo. A cada cabecera se le agrega entonces una fracción de dichas inversiones, siendo dicha proporción el ratio entre su contribución a la carga del anillo (e1hosttandemTOTALx) y la carga total del anillo (e1compHTTOTALx) *)

CosteExtra2FHTx[[AnilloCabeceras[[ip]]]] = $\frac{\text{costeTerminacionesFibrax}[[\text{AnilloCabeceras}[[ip]]]] * e1hosttandemTOTALx[[AnilloCabeceras[[ip]]]]}{e1compHTTOTALx[[AnilloCabeceras[[ip]]]]};$

(* A nivel HT todos los costos deben ser imputados a las cabeceras no Tandem. En ese sentido, cuando hay anillo HT, los costos de las terminaciones de cada una de las Tandem deben ser repartidos entre todas las centrales cabeceras que forman parte del anillo. A cada cabecera se le agrega entonces una fracción de dichas inversiones, siendo dicha proporción el ratio entre su contribución a la carga del anillo e1hosttandemTOTALx y la carga total del anillo (e1compHTTOTALx) *)

, {ip, np}];

fiberinvhosttandemx1 = zerox;

fiberinvhosttandemalqx1A1 = zerox;

fiberinvhosttandemalqx1B1 = zerox;

fiberinvhosttandemalqx1C1 = zerox;

fiberinvhosttandemalqx1LL = zerox;

fiberinvhosttandemalqx1LP = zerox;

fiberinvhosttandemalqx1AS = zerox;

fiberinvhosttandemalqx1BS = zerox;

fiberinvhosttandemalqx1CS = zerox;

fiberinvhosttandemalqx1ABC2 = zerox;

fiberinvhosttandemalqx1ABCnop = zerox;

fiberinvhosttandemitxx1 = zerox;

fiberinvhosttandemadslx1 = zerox;

fiberinvhosttandemTOTALx1 = zerox;

Do[If[e1compHTTOTALx[[ix]] ≠ 0 ∧ tandem1x[[ix]] ≠ 1,

fiberinvhosttandemTOTALx1[[ix]] = If[esAnillodx[[ix]] ≠ NA, 1, 2] * CosteFijoFHTx[[ix]] + If[esAnillodx[[ix]] ≠ NA, 1, 2] *
 CosteTributariosFHTx[[ix]] numtributariosFHTx[[ix]] + CosteFijoFHTx[[ix]] * repfibra2x[[ix]] + CosteExtraFHTx[[ix]];

(* Se estiman los costos asociados a los equipos: Equipo de Tx, Tributarios, Repetidora y Costo Extra. Notese que en el caso del equipo de Tx y el tributario si la central cabecera es parte de un anillo sólo considera una unidad de dichos elementos, mientras que si no son parte del anillo departamental considera dos unidades de cada uno de ellos. Se asegura que a nivel de centrales Tandem el resultado debe ser siempre cero *)

fiberinvhosttandemx1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] * $\frac{e1compHTx[[ix]]}{e1compHTTOTALx[[ix]]};$

fiberinvhosttandemalqx1A1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] * $\frac{e1compalqHTxA1[[ix]]}{e1compHTTOTALx[[ix]]};$

fiberinvhosttandemalqx1B1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] * $\frac{e1compalqHTxB1[[ix]]}{e1compHTTOTALx[[ix]]};$

fiberinvhosttandemalqx1C1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] * $\frac{e1compalqHTxC1[[ix]]}{e1compHTTOTALx[[ix]]};$

```

fiberinvhosttandemalqx1LL[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxLL[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1LP[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxLP[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1AS[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxAS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1BS[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxBS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1CS[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxCS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1ABC2[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxABC2[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemalqx1ABCnop[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compalqHTxABCnop[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemitxx1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compitxHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
fiberinvhosttandemadslx1[[ix]] = fiberinvhosttandemTOTALx1[[ix]] *  $\frac{e1compadslHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
];
, {ix, nx}];

(* Luego de sumar los costos, se identifica el costo imputable a cada servicio,
para lo cual se multiplica la inversión total por la fracción de asignación establecida de cada servicio *)

Print[TableForm[
  Table[{ix, idx[[ix]], y[[ix, 13]], repfibra2x[[ix]], numtributariosFHTx[[ix]], VelocidadFHTx[[ix]], CosteFijoFHTx[[ix]],
    CosteTributariosFHTx[[ix]], CosteExtraFHTx[[ix]], fiberinvhosttandemx1[[ix]], fiberinvhosttandemalqx1A1[[ix]],
    fiberinvhosttandemalqx1B1[[ix]], fiberinvhosttandemalqx1C1[[ix]], fiberinvhosttandemalqx1LL[[ix]],
    fiberinvhosttandemalqx1LP[[ix]], fiberinvhosttandemalqx1AS[[ix]], fiberinvhosttandemalqx1BS[[ix]],
    fiberinvhosttandemalqx1CS[[ix]], fiberinvhosttandemalqx1ABC2[[ix]], fiberinvhosttandemalqx1ABCnop[[ix]],
    fiberinvhosttandemadslx1[[ix]], fiberinvhosttandemitxx1[[ix]], fiberinvhosttandemTOTALx1[[ix]]}, {ix, nx}],
  TableDirections -> {Column, Row}, TableSpacing -> {0, 1}, TableHeadings ->
  {None, {"ix", "idx", "Nombre Central", "repfibra2x", "numtributariosFHTx", "VelocidadFHTx", "CosteFijoFHTx",
    "CosteTributariosFHTx", "CosteExtraFHTx", "fiberinvhosttandemx1", "fiberinvhosttandemalqx1A1",
    "fiberinvhosttandemalqx1B1", "fiberinvhosttandemalqx1C1", "fiberinvhosttandemalqx1LL",
    "fiberinvhosttandemalqx1LP", "fiberinvhosttandemalqx1AS", "fiberinvhosttandemalqx1BS",
    "fiberinvhosttandemalqx1CS", "fiberinvhosttandemalqx1ABC2", "fiberinvhosttandemalqx1ABCnop",
    "fiberinvhosttandemadslx1", "fiberinvhosttandemitxx1", "fiberinvhosttandemTOTALx1"}}]];

fiberinvhosttandemx2 = zerox;
fiberinvhosttandemalqx2A1 = zerox;
fiberinvhosttandemalqx2B1 = zerox;
fiberinvhosttandemalqx2C1 = zerox;
fiberinvhosttandemalqx2LL = zerox;
fiberinvhosttandemalqx2LP = zerox;
fiberinvhosttandemalqx2AS = zerox;
fiberinvhosttandemalqx2BS = zerox;
fiberinvhosttandemalqx2CS = zerox;
fiberinvhosttandemalqx2ABC2 = zerox;
fiberinvhosttandemalqx2ABCnop = zerox;
fiberinvhosttandemitxx2 = zerox;
fiberinvhosttandemadslx2 = zerox;
fiberinvhosttandemTOTALx2 = zerox;
Do[If[e1compHTTtotalx[[ix]] != 0 & tandem1x[[ix]] != 1,
  fiberinvhosttandemTOTALx2[[ix]] = If[transtechx[[ix]] == Urbana, costedelafibraurbKm, If[transtechx[[ix]] == Enterrado,
    TxFibraCosteCableKm, If[transtechx[[ix]] == Aereo, TxFibraCosteCableAereoKm, 0]]] *
    disttandemx[[ix]] + costeTerminacionesFibrax[[ix]] + CosteExtra2FHTx[[ix]];
  (* Se estiman los costos asociados a los cables propiamente dichos (Costo por Km * Distancia en Kms) y
  las terminaciones de Fibra. Nótese que si la central usa Fibra canalizada el costo del cable es particular,
  para todos los demás casos es un precio estándar. Se asegura que a nivel de centrales Tandem el resultado
  debe ser siempre cero *) fiberinvhosttandemx2[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2A1[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxA1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2B1[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxB1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2C1[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxC1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2LL[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxLL[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2LP[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxLP[[ix]]}{e1compHTTtotalx[[ix]]}$ ;
  fiberinvhosttandemalqx2AS[[ix]] = fiberinvhosttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxAS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

```

```

fiberinvhostttandemalqx2BS[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxBS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

fiberinvhostttandemalqx2CS[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxCS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

fiberinvhostttandemalqx2ABC2[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxABC2[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

fiberinvhostttandemalqx2ABCnop[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compalqHTxABCnop[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

fiberinvhostttandemitxx2[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compitxHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

fiberinvhostttandemadslx2[[ix]] = fiberinvhostttandemTOTALx2[[ix]] *  $\frac{e1compadslHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;];

, {ix, nx}];

(* Luego, se identifica el costo imputable a cada servicio,
para lo cual se multiplica la inversión total por la fracción de asignación establecida. *)

Print[TableForm[
  Table[{ix, idx[[ix]], y[[ix, 13]], costeTerminacionesFibrax[[ix]], CosteExtra2FHTx[[ix]], fiberinvhostttandemx2[[ix]],
    fiberinvhostttandemalqx2A1[[ix]], fiberinvhostttandemalqx2B1[[ix]], fiberinvhostttandemalqx2C1[[ix]],
    fiberinvhostttandemalqx2LL[[ix]], fiberinvhostttandemalqx2LP[[ix]], fiberinvhostttandemalqx2AS[[ix]],
    fiberinvhostttandemalqx2BS[[ix]], fiberinvhostttandemalqx2CS[[ix]], fiberinvhostttandemalqx2ABC2[[ix]],
    fiberinvhostttandemalqx2ABCnop[[ix]], fiberinvhostttandemadslx2[[ix]], fiberinvhostttandemitxx2[[ix]],
    fiberinvhostttandemTOTALx2[[ix]]}, {ix, nx}], TableDirections -> {Column, Row}, TableSpacing -> {0, 1},
  TableHeadings -> {None, {"ix", "idx", "Nombre Central", "costeTerminacionesFibrax",
    "CosteExtra2FHTx", "fiberinvhostttandemx2", "fiberinvhostttandemalqx2A1",
    "fiberinvhostttandemalqx2B1", "fiberinvhostttandemalqx2C1", "fiberinvhostttandemalqx2LL",
    "fiberinvhostttandemalqx2LP", "fiberinvhostttandemalqx2AS", "fiberinvhostttandemalqx2BS",
    "fiberinvhostttandemalqx2CS", "fiberinvhostttandemalqx2ABC2", "fiberinvhostttandemalqx2ABCnop",
    "fiberinvhostttandemadslx2", "fiberinvhostttandemitxx2", "fiberinvhostttandemTOTALx2"}}]];

fiberinvhostttandemx3 = zerox;
fiberinvhostttandemalqx3A1 = zerox;
fiberinvhostttandemalqx3B1 = zerox;
fiberinvhostttandemalqx3C1 = zerox;
fiberinvhostttandemalqx3LL = zerox;
fiberinvhostttandemalqx3LP = zerox;
fiberinvhostttandemalqx3AS = zerox;
fiberinvhostttandemalqx3BS = zerox;
fiberinvhostttandemalqx3CS = zerox;
fiberinvhostttandemalqx3ABC2 = zerox;
fiberinvhostttandemalqx3ABCnop = zerox;
fiberinvhostttandemitxx3 = zerox;
fiberinvhostttandemadslx3 = zerox;
fiberinvhostttandemTOTALx3 = zerox;
Do[
  If[e1compHTTtotalx[[ix]] != 0 & tandem1x[[ix]] != 1, fiberinvhostttandemTOTALx3[[ix]] = CostePorKmFHTx[[ix]] disttandemx[[ix]];
  (* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia).Se
  asegura que a nivel de centrales Tandem el resultado debe ser siempre cero *)

  fiberinvhostttandemx3[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3A1[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxA1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3B1[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxB1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3C1[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxC1[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3LL[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxLL[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3LP[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxLP[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3AS[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxAS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3BS[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxBS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3CS[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxCS[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3ABC2[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxABC2[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemalqx3ABCnop[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compalqHTxABCnop[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

  fiberinvhostttandemitxx3[[ix]] = fiberinvhostttandemTOTALx3[[ix]] *  $\frac{e1compitxHTx[[ix]]}{e1compHTTtotalx[[ix]]}$ ;

```



```

    fiberinvhosttandemadslx3[[ix]] = fiberinvhosttandemTOTALx3[[ix]] *  $\frac{e1compadslHTx[[ix]]}{e1compHTTOTALx[[ix]]}$ ;
    , {ix, nx}]];

(* Luego se identifica el costo imputable a cada servicio,
para lo cual se multiplica la inversión total por la fracción de asignación establecida. *)

Print[TableForm[
  Table[{ix, idx[[ix]], y[[ix, 13]], CostePorKmFHTx[[ix]], fiberinvhosttandemx3[[ix]], fiberinvhosttandemalqx3A1[[ix]],
    fiberinvhosttandemalqx3B1[[ix]], fiberinvhosttandemalqx3C1[[ix]], fiberinvhosttandemalqx3LL[[ix]],
    fiberinvhosttandemalqx3LP[[ix]], fiberinvhosttandemalqx3AS[[ix]], fiberinvhosttandemalqx3BS[[ix]],
    fiberinvhosttandemalqx3CS[[ix]], fiberinvhosttandemalqx3ABC2[[ix]], fiberinvhosttandemalqx3ABCnop[[ix]],
    fiberinvhosttandemadslx3[[ix]], fiberinvhosttandemitxx3[[ix]], fiberinvhosttandemTOTALx3[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"ix", "idx", "Nombre Central", "CostePorKmFHTx", "fiberinvhosttandemx3", "fiberinvhosttandemalqx3A1",
      "fiberinvhosttandemalqx3B1", "fiberinvhosttandemalqx3C1", "fiberinvhosttandemalqx3LL",
      "fiberinvhosttandemalqx3LP", "fiberinvhosttandemalqx3AS", "fiberinvhosttandemalqx3BS",
      "fiberinvhosttandemalqx3CS", "fiberinvhosttandemalqx3ABC2", "fiberinvhosttandemalqx3ABCnop",
      "fiberinvhosttandemadslx3", "fiberinvhosttandemitxx3", "fiberinvhosttandemTOTALx3"}}]];

fiberinvhosttandemx = fiberinvhosttandemx1 + fiberinvhosttandemx2 + fiberinvhosttandemx3;
fiberinvhosttandemalqx1A1 = fiberinvhosttandemalqx1A1 + fiberinvhosttandemalqx2A1 + fiberinvhosttandemalqx3A1;
fiberinvhosttandemalqx1B1 = fiberinvhosttandemalqx1B1 + fiberinvhosttandemalqx2B1 + fiberinvhosttandemalqx3B1;
fiberinvhosttandemalqx1C1 = fiberinvhosttandemalqx1C1 + fiberinvhosttandemalqx2C1 + fiberinvhosttandemalqx3C1;
fiberinvhosttandemalqx1LL = fiberinvhosttandemalqx1LL + fiberinvhosttandemalqx2LL + fiberinvhosttandemalqx3LL;
fiberinvhosttandemalqx1LP = fiberinvhosttandemalqx1LP + fiberinvhosttandemalqx2LP + fiberinvhosttandemalqx3LP;
fiberinvhosttandemalqx1AS = fiberinvhosttandemalqx1AS + fiberinvhosttandemalqx2AS + fiberinvhosttandemalqx3AS;
fiberinvhosttandemalqx1BS = fiberinvhosttandemalqx1BS + fiberinvhosttandemalqx2BS + fiberinvhosttandemalqx3BS;
fiberinvhosttandemalqx1CS = fiberinvhosttandemalqx1CS + fiberinvhosttandemalqx2CS + fiberinvhosttandemalqx3CS;
fiberinvhosttandemalqxABC2 =
  fiberinvhosttandemalqx1ABC2 + fiberinvhosttandemalqx2ABC2 + fiberinvhosttandemalqx3ABC2;
fiberinvhosttandemalqxABCnop = fiberinvhosttandemalqx1ABCnop +
  fiberinvhosttandemalqx2ABCnop + fiberinvhosttandemalqx3ABCnop;
fiberinvhosttandemitxx = fiberinvhosttandemitxx1 + fiberinvhosttandemitxx2 + fiberinvhosttandemitxx3;
fiberinvhosttandemadslx = fiberinvhosttandemadslx1 + fiberinvhosttandemadslx2 + fiberinvhosttandemadslx3;
fiberinvhosttandemTOTALx = fiberinvhosttandemTOTALx1 + fiberinvhosttandemTOTALx2 + fiberinvhosttandemTOTALx3;

Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], fiberinvhosttandemx[[ix]], fiberinvhosttandemalqx1A1[[ix]],
  fiberinvhosttandemalqx1B1[[ix]], fiberinvhosttandemalqx1C1[[ix]], fiberinvhosttandemalqx1LL[[ix]],
  fiberinvhosttandemalqx1LP[[ix]], fiberinvhosttandemalqx1AS[[ix]], fiberinvhosttandemalqx1BS[[ix]],
  fiberinvhosttandemalqx1CS[[ix]], fiberinvhosttandemalqx1ABC2[[ix]], fiberinvhosttandemalqx1ABCnop[[ix]],
  fiberinvhosttandemadslx[[ix]], fiberinvhosttandemitxx[[ix]], fiberinvhosttandemTOTALx[[ix]]}, {ix, nx}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"ix", "idx", "Nombre Central", "fiberinvhosttandemx", "fiberinvhosttandemalqx1A1",
      "fiberinvhosttandemalqx1B1", "fiberinvhosttandemalqx1C1", "fiberinvhosttandemalqx1LL",
      "fiberinvhosttandemalqx1LP", "fiberinvhosttandemalqx1AS", "fiberinvhosttandemalqx1BS",
      "fiberinvhosttandemalqx1CS", "fiberinvhosttandemalqx1ABC2", "fiberinvhosttandemalqx1ABCnop",
      "fiberinvhosttandemadslx", "fiberinvhosttandemitxx", "fiberinvhosttandemTOTALx"}}]];

```

```

Print[Style["ESTIMACIÓN DE INVERSIONES EN FIBRA A NIVEL REMOTA-CABECERA",
  16, Bold, White, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

numtributariosFRHx = Ceiling[ $\frac{e1remotehostTOTALx}{21}$ ];

(* Identifica el número de tributarios por central *)

VelocidadFRHx = zerox;
CostePorKmFRHx = zerox;
CosteFijoFRHx = zerox;
CosteTributariosFRHx = zerox;
Do[If[e1compRHTOTALx[[ix]] ≠ 0,
  Do[If[e1compRHTOTALx[[ix]] > TxFibraLimites[[i]], VelocidadFRHx[[ix]] = i];, {i, Length[TxFibraLimites]}];
  If[VelocidadFRHx[[ix]] < Length[TxFibraLimites], VelocidadFRHx[[ix]] = VelocidadFRHx[[ix]] + 1]];

(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)
If[transtechx[[ix]] == Urbana, CostePorKmFRHx[[ix]] = TxFibraCosteKmUrban;, If[transtechx[[ix]] == Enterrado,
  CostePorKmFRHx[[ix]] = TxFibraCosteKmEnterrado;, CostePorKmFRHx[[ix]] = TxFibraCosteKmAereo;];];

(* Especifica el costo asociado a la obra civil dependiendo del tipo de Fibra *)
If[VelocidadFRHx[[ix]] ≠ 0, CosteFijoFRHx[[ix]] = TxFibraCosteFijo[VelocidadFRHx[[ix]]];
  (* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)
  CosteTributariosFRHx[[ix]] = TxFibraCosteTributario[VelocidadFRHx[[ix]]];];, {ix, nx}];

(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

CosteExtraFRHx = zerox;
CosteExtra2FRHx = zerox;
Do[If[esAnillopx[[ix]] == 1 ∧ e1AnilloTOTALp[provx[[ix]]] ≠ 0, CosteExtraFRHx[[ix]] =

```

```

    (CosteFijoFRHx[[ix]] + CosteTributariosFRHx[[ix]] * Ceiling[ $\frac{e1AnilloTOTALp[[provx[[ix]]]]}{21}$ ]) *  $\frac{e1remotehostTOTALx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    (* A nivel RH todos los costos deben ser imputados a las remotas. En ese sentido, cuando hay anillo RH,
    los costos de los equipos de Tx y de los tributarios de cada una de las Cabeceras deben ser repartidos
    entre todas las centrales remotas que forman parte del anillo. A cada remota se le agrega entonces
    una fracción de dichas inversiones, siendo dicha proporción el ratio entre su contribución
    a la carga del anillo (e1remotehostTOTALx) y la carga total del anillo (e1compFRHTOTALx) *)

    CosteExtra2FRHx[[ix]] = costeTerminacionesFibrax[[ix]] *  $\frac{e1remotehostTOTALx[[ix]]}{e1compRHTOTALx[[ix]]}$  * 0;

    (* A nivel RH todos los costos deben ser imputados a las remotas. En ese sentido, cuando hay anillo RH,
    los costos de las terminaciones de cada una de las Cabeceras deben ser repartidos entre todas
    las centrales remotas que forman parte del anillo. A cada remota se le agrega entonces una
    fracción de dichas inversiones, siendo dicha proporción el ratio entre su contribución a
    la carga del anillo (e1remotehostTOTALx) y la carga total del anillo (e1compRHTOTALx) *)];

    , {ix, nx}];

```

```

fiberinvremotehostx1 = zerox;
fiberinvremotehostalqx1A1 = zerox;
fiberinvremotehostalqx1B1 = zerox;
fiberinvremotehostalqx1C1 = zerox;
fiberinvremotehostalqx1LL = zerox;
fiberinvremotehostalqx1LP = zerox;
fiberinvremotehostalqx1AS = zerox;
fiberinvremotehostalqx1BS = zerox;
fiberinvremotehostalqx1CS = zerox;
fiberinvremotehostalqx1ABC2 = zerox;
fiberinvremotehostalqx1ABCnop = zerox;
fiberinvremotehostitxx1 = zerox;
fiberinvremotehostadslx1 = zerox;
fiberinvremotehostTOTALx1 = zerox;

```

```

Do[If[e1compRHTOTALx[[ix]] ≠ 0 ∧ host1x[[ix]] ≠ 1,
    fiberinvremotehostTOTALx1[[ix]] = If[esAnillopx[[ix]] == 1, 1, 2] * CosteFijoFRHx[[ix]] + If[esAnillopx[[ix]] == 1, 1, 2] *
    CosteTributariosFRHx[[ix]] numtributariosFRHx[[ix]] + CosteFijoFRHx[[ix]] repfibra2x[[ix]] + CosteExtraFRHx[[ix]];

    (* Se estiman los costos asociados a los equipos: Equipo de Tx, Tributarios,
    Repetidora y Costo Extra. Notese que en el caso del equipo de Tx y el tributario si la
    central remota es parte de un anillo sólo considera una unidad de dichos elementos,
    mientras que si no son parte del anillo provincial considera dos unidades de cada uno de
    ellos. Se asegura que a nivel de centrales Host el resultado debe ser siempre cero *)

    fiberinvremotehostx1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1A1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxA1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1B1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxB1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1C1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxC1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1LL[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxLL[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1LP[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxLP[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1AS[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxAS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1BS[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxBS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1CS[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxCS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1ABC2[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxABC2[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx1ABCnop[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compalqRHxABCnop[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostitxx1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compitxRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostadslx1[[ix]] = fiberinvremotehostTOTALx1[[ix]] *  $\frac{e1compadslRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ];

    , {ix, nx}];

```

(* Luego se identifica el costo imputable a cada servicio,
para lo cual se multiplica la inversión total por la fracción de asignación establecida *)

```

Print[TableForm[
    Table[{ix, idx[[ix]], y[[ix, 13]], repfibra2x[[ix]], numtributariosFRHx[[ix]], VelocidadFRHx[[ix]], CosteFijoFRHx[[ix]],
    CosteTributariosFRHx[[ix]], CosteExtraFRHx[[ix]], fiberinvremotehostx1[[ix]], fiberinvremotehostalqx1A1[[ix]],
    fiberinvremotehostalqx1B1[[ix]], fiberinvremotehostalqx1C1[[ix]], fiberinvremotehostalqx1LL[[ix]],
    fiberinvremotehostalqx1LP[[ix]], fiberinvremotehostalqx1AS[[ix]], fiberinvremotehostalqx1BS[[ix]],
    fiberinvremotehostalqx1CS[[ix]], fiberinvremotehostalqx1ABC2[[ix]], fiberinvremotehostalqx1ABCnop[[ix]],

```



```

    fiberinvremotehostadslx1[[ix]], fiberinvremotehostitxx1[[ix]], fiberinvremotehostTOTALx1[[ix]]}, {ix, nx}},
    TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"ix", "idx", "Nombre Central", "repfibra2x", "numtributariosFRHx", "VelocidadFRHx", "CosteFijoFRHx",
    "CosteTributariosFRHx", "CosteExtraFRHx", "fiberinvremotehostx1", "fiberinvremotehostalqx1A1",
    "fiberinvremotehostalqx1B1", "fiberinvremotehostalqx1C1", "fiberinvremotehostalqx1LL",
    "fiberinvremotehostalqx1LP", "fiberinvremotehostalqx1AS", "fiberinvremotehostalqx1BS",
    "fiberinvremotehostalqx1CS", "fiberinvremotehostalqx1ABC2", "fiberinvremotehostalqx1ABCnop",
    "fiberinvremotehostadslx1", "fiberinvremotehostitxx1", "fiberinvremotehostTOTALx1"}}}];

fiberinvremotehostx2 = zerox;
fiberinvremotehostalqx2A1 = zerox;
fiberinvremotehostalqx2B1 = zerox;
fiberinvremotehostalqx2C1 = zerox;
fiberinvremotehostalqx2LL = zerox;
fiberinvremotehostalqx2LP = zerox;
fiberinvremotehostalqx2AS = zerox;
fiberinvremotehostalqx2BS = zerox;
fiberinvremotehostalqx2CS = zerox;
fiberinvremotehostalqx2ABC2 = zerox;
fiberinvremotehostalqx2ABCnop = zerox;
fiberinvremotehostitxx2 = zerox;
fiberinvremotehostadslx2 = zerox;
fiberinvremotehostTOTALx2 = zerox;
Do[If[e1compRHTOTALx[[ix]] ≠ 0 ∧ host1x[[ix]] ≠ 1, fiberinvremotehostTOTALx2[[ix]] =
    If[transtechx[[ix]] == Urbana, costedelafibraurbKm, If[transtechx[[ix]] == Enterrado, TxFibraCosteCableKm, 0]]
    disthostx[[ix]] + costeTerminacionesFibrax[[ix]] + CosteExtra2FRHx[[ix]];
    (* Se estiman los costos asociados a los cables propiamente dichos (Costo por Km * Distancia en Kms) y
    las terminaciones de Fibra. Nótese que si la central usa Fibra canalizada el costo del cable es particular,
    para todos los demás casos es un precio estándar. Se asegura que a nivel
    de centrales Host el resultado debe ser siempre cero *)

    fiberinvremotehostx2[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2A1[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxA1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2B1[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxB1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2C1[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxC1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2LL[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxLL[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2LP[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxLP[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2AS[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxAS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2BS[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxBS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2CS[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxCS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2ABC2[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxABC2[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostalqx2ABCnop[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compalqRHxABCnop[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostitxx2[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compitxRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

    fiberinvremotehostadslx2[[ix]] = fiberinvremotehostTOTALx2[[ix]] *  $\frac{e1compadslRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ];

    , {ix, nx}];

    (* Luego se identifica el costo imputable a cada servicio,
    para lo cual se multiplica la inversión total por la fracción de asignación establecida. *)

Print[TableForm[
    Table[{ix, idx[[ix]], y[[ix, 13]], costeTerminacionesFibrax[[ix]], CosteExtra2FRHx[[ix]], fiberinvremotehostx2[[ix]],
    fiberinvremotehostalqx2A1[[ix]], fiberinvremotehostalqx2B1[[ix]], fiberinvremotehostalqx2C1[[ix]],
    fiberinvremotehostalqx2LL[[ix]], fiberinvremotehostalqx2LP[[ix]], fiberinvremotehostalqx2AS[[ix]],
    fiberinvremotehostalqx2BS[[ix]], fiberinvremotehostalqx2CS[[ix]], fiberinvremotehostalqx2ABC2[[ix]],
    fiberinvremotehostalqx2ABCnop[[ix]], fiberinvremotehostadslx2[[ix]], fiberinvremotehostitxx2[[ix]],
    fiberinvremotehostTOTALx2[[ix]]}, {ix, nx}], TableDirections → {Column, Row}, TableSpacing → {0, 1},
    TableHeadings → {None, {"ix", "idx", "Nombre Central", "costeTerminacionesFibrax",
    "CosteExtra2FRHx", "fiberinvremotehostx2", "fiberinvremotehostalqx2A1",
    "fiberinvremotehostalqx2B1", "fiberinvremotehostalqx2C1", "fiberinvremotehostalqx2LL",
    "fiberinvremotehostalqx2LP", "fiberinvremotehostalqx2AS", "fiberinvremotehostalqx2BS",
    "fiberinvremotehostalqx2CS", "fiberinvremotehostalqx2ABC2", "fiberinvremotehostalqx2ABCnop",
    "fiberinvremotehostadslx2", "fiberinvremotehostitxx2", "fiberinvremotehostTOTALx2"}}}];

fiberinvremotehostx3 = zerox;
fiberinvremotehostalqx3A1 = zerox;

```

```

fiberinvremotehostalqx3B1 = zerox;
fiberinvremotehostalqx3C1 = zerox;
fiberinvremotehostalqx3LL = zerox;
fiberinvremotehostalqx3LP = zerox;
fiberinvremotehostalqx3ABC2 = zerox;
fiberinvremotehostalqx3AS = zerox;
fiberinvremotehostalqx3BS = zerox;
fiberinvremotehostalqx3CS = zerox;
fiberinvremotehostalqx3ABCnop = zerox;
fiberinvremotehostitxx3 = zerox;
fiberinvremotehostadslx3 = zerox;
fiberinvremotehostTOTALx3 = zerox;

Do[If[e1compRHTOTALx[[ix]] ≠ 0 ∧ host1x[[ix]] ≠ 1, fiberinvremotehostTOTALx3[[ix]] = CostePorKmFRHx[[ix]] disthostx[[ix]];

(* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia). Se
asegura que a nivel de centrales Host el resultado debe ser siempre cero *)

fiberinvremotehostx3[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3A1[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxA1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3B1[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxB1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3C1[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxC1[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3LL[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxLL[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3LP[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxLP[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3ABC2[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxABC2[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3AS[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxAS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3BS[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxBS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3CS[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxCS[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostalqx3ABCnop[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compalqRHxABCnop[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostitxx3[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compitxRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ;

fiberinvremotehostadslx3[[ix]] = fiberinvremotehostTOTALx3[[ix]] *  $\frac{e1compadslRHx[[ix]]}{e1compRHTOTALx[[ix]]}$ ];

, {ix, nx}];

(* Luego se identifica el costo imputable a cada servicio,
para lo cual se multiplica la inversión total por la fracción de asignación establecida. *)

fiberinvremotehostx = fiberinvremotehostx1 + fiberinvremotehostx2 + fiberinvremotehostx3;
fiberinvremotehostalqxA1 = fiberinvremotehostalqx1A1 + fiberinvremotehostalqx2A1 + fiberinvremotehostalqx3A1;
fiberinvremotehostalqxB1 = fiberinvremotehostalqx1B1 + fiberinvremotehostalqx2B1 + fiberinvremotehostalqx3B1;
fiberinvremotehostalqxC1 = fiberinvremotehostalqx1C1 + fiberinvremotehostalqx2C1 + fiberinvremotehostalqx3C1;
fiberinvremotehostalqxLL = fiberinvremotehostalqx1LL + fiberinvremotehostalqx2LL + fiberinvremotehostalqx3LL;
fiberinvremotehostalqxLP = fiberinvremotehostalqx1LP + fiberinvremotehostalqx2LP + fiberinvremotehostalqx3LP;
fiberinvremotehostalqxAS = fiberinvremotehostalqx1AS + fiberinvremotehostalqx2AS + fiberinvremotehostalqx3AS;
fiberinvremotehostalqxBs = fiberinvremotehostalqx1BS + fiberinvremotehostalqx2BS + fiberinvremotehostalqx3BS;
fiberinvremotehostalqxCS = fiberinvremotehostalqx1CS + fiberinvremotehostalqx2CS + fiberinvremotehostalqx3CS;
fiberinvremotehostalqxABC2 =
  fiberinvremotehostalqx1ABC2 + fiberinvremotehostalqx2ABC2 + fiberinvremotehostalqx3ABC2;
fiberinvremotehostalqxABCnop = fiberinvremotehostalqx1ABCnop +
  fiberinvremotehostalqx2ABCnop + fiberinvremotehostalqx3ABCnop;
fiberinvremotehostitxx = fiberinvremotehostitxx1 + fiberinvremotehostitxx2 + fiberinvremotehostitxx3;
fiberinvremotehostadslx = fiberinvremotehostadslx1 + fiberinvremotehostadslx2 + fiberinvremotehostadslx3;
fiberinvremotehostTOTALx = fiberinvremotehostTOTALx1 + fiberinvremotehostTOTALx2 + fiberinvremotehostTOTALx3;

Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], fiberinvremotehostx[[ix]], fiberinvremotehostalqxA1[[ix]],
  fiberinvremotehostalqxB1[[ix]], fiberinvremotehostalqxC1[[ix]], fiberinvremotehostalqxLL[[ix]],
  fiberinvremotehostalqxLP[[ix]], fiberinvremotehostalqxAS[[ix]], fiberinvremotehostalqxBs[[ix]],
  fiberinvremotehostalqxCS[[ix]], fiberinvremotehostalqxABC2[[ix]], fiberinvremotehostalqxABCnop[[ix]],
  fiberinvremotehostadslx[[ix]], fiberinvremotehostitxx[[ix]], fiberinvremotehostTOTALx[[ix]]}, {ix, nx}],
TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
{None, {"ix", "idx", "Nombre Central", "fiberinvremotehostx", "fiberinvremotehostalqxA1",
  "fiberinvremotehostalqxB1", "fiberinvremotehostalqxC1", "fiberinvremotehostalqxLL",
  "fiberinvremotehostalqxLP", "fiberinvremotehostalqxAS", "fiberinvremotehostalqxBs",
  "fiberinvremotehostalqxCS", "fiberinvremotehostalqxABC2", "fiberinvremotehostalqxABCnop",
  "fiberinvremotehostadslx", "fiberinvremotehostitxx", "fiberinvremotehostTOTALx"}}]];

```

INVERSIÓN COMPILADA

```
Print[Style["INVERSIÓN COMPILADA A NIVEL CABECERA-TANDEM",
```

```
16, Bold, White, Underlined, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
```

```
CosteAñadidoTrxPorDistanciaHTx = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxA1 = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxB1 = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxC1 = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxLL = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxLP = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxAS = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxBS = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxCS = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxABC2 = zerox;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxABCnop = zerox;
```

```
CosteAñadidoTrxPorDistanciaaitxHTx = zerox;
```

```
CosteAñadidoTrxPorDistanciaaadsLHTx = zerox;
```

```
CosteAñadidoTrxPorDistanciaHTTTOTALx = zerox;
```

```
Do[If[deptx[[ix]] ≠ LIMA ∧ host1x[[ix]] == 1 ∧ e1compHTTTOTALx[[ix]] ≠ 0 ∧ transtechx[[ix]] ≠ Radio,
```

```
CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]] = AñadidoTrxPorKm distance[ix, idinvertx[CentralRef]];
```

```
CosteAñadidoTrxPorDistanciaHTx[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compHTx[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxA1[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxA1[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxB1[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxB1[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxC1[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxC1[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxLL[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxLL[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxLP[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxLP[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxAS[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxAS[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxBS[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxBS[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxCS[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxCS[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxABC2[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxABC2[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaalqHTxABCnop[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compaalqHTxABCnop[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaitxHTx[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compitxHTx[[ix]]}{e1compHTTTOTALx[[ix]]}$ ;
```

```
CosteAñadidoTrxPorDistanciaaadsLHTx[[ix]] = CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]  $\frac{e1compadsLHTx[[ix]]}{e1compHTTTOTALx[[ix]]}$ ];
```

```
, {ix, nx}];
```

```
(* El costo de traslado de los equipos de Tx a
```

```
provincias: Si la central no esta en Lima y es remota equivale a la distancia de cada central
hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En
todas las Host entonces el valor es cero. El resultado se multiplica por el respectivo factor
de imputación para estimar el costo de cada servicio. Nótese que si el número de E1s es cero,
entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo,
en todas las remotas e1compHTTTOTALx=0,
entonces costo añadido será cero. Conclusión,
a nivel HT todo es cero *)
```

```
Print[TableForm[
```

```
Table[{ix, idx[[ix]], y[[ix, 13]], CosteAñadidoTrxPorDistanciaHTx[[ix]], CosteAñadidoTrxPorDistanciaaalqHTxA1[[ix]],
CosteAñadidoTrxPorDistanciaaalqHTxB1[[ix]], CosteAñadidoTrxPorDistanciaaalqHTxC1[[ix]],
CosteAñadidoTrxPorDistanciaaalqHTxLL[[ix]], CosteAñadidoTrxPorDistanciaaalqHTxLP[[ix]],
CosteAñadidoTrxPorDistanciaaalqHTxAS[[ix]], CosteAñadidoTrxPorDistanciaaalqHTxBS[[ix]],
CosteAñadidoTrxPorDistanciaaalqHTxCS[[ix]], CosteAñadidoTrxPorDistanciaaalqHTxABC2[[ix]],
CosteAñadidoTrxPorDistanciaaalqHTxABCnop[[ix]], CosteAñadidoTrxPorDistanciaaadsLHTx[[ix]],
CosteAñadidoTrxPorDistanciaaitxHTx[[ix]], CosteAñadidoTrxPorDistanciaHTTTOTALx[[ix]]}, {ix, nx}],
```

```
TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
```

```
{None, {"ix", "idx", "Nombre Central", "CosteAñadidoTrxPorDistanciaHTx", "CosteAñadidoTrxPorDistanciaaalqHTxA1",
"CosteAñadidoTrxPorDistanciaaalqHTxB1", "CosteAñadidoTrxPorDistanciaaalqHTxC1",
"CosteAñadidoTrxPorDistanciaaalqHTxLL", "CosteAñadidoTrxPorDistanciaaalqHTxLP",
"CosteAñadidoTrxPorDistanciaaalqHTxAS", "CosteAñadidoTrxPorDistanciaaalqHTxBS",
"CosteAñadidoTrxPorDistanciaaalqHTxCS", "CosteAñadidoTrxPorDistanciaaalqHTxABC2",
"CosteAñadidoTrxPorDistanciaaalqHTxABCnop", "CosteAñadidoTrxPorDistanciaaadsLHTx",
"CosteAñadidoTrxPorDistanciaaitxHTx", "CosteAñadidoTrxPorDistanciaHTTTOTALx"}}];
```

```

invhosttandemx1 = Table[If[satx[[ix]], 0, radioinvhosttandemx[[ix]], {ix, nx}];
invhosttandemLocx1 = zerox;
invhosttandemLDNx1 = zerox;
invhosttandemLDIx1 = zerox;
invhosttandemalqx1A1 = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1A1[[ix]], {ix, nx}];
invhosttandemalqx1B1 = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1B1[[ix]], {ix, nx}];
invhosttandemalqx1C1 = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1C1[[ix]], {ix, nx}];
invhosttandemalqx1LL = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1LL[[ix]], {ix, nx}];
invhosttandemalqx1LP = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1LP[[ix]], {ix, nx}];
invhosttandemalqx1AS = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1AS[[ix]], {ix, nx}];
invhosttandemalqx1BS = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1BS[[ix]], {ix, nx}];
invhosttandemalqx1CS = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1CS[[ix]], {ix, nx}];
invhosttandemalqx1ABC2 = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1ABC2[[ix]], {ix, nx}];
invhosttandemalqx1ABCnop = Table[If[satx[[ix]], 0, radioinvhosttandemalqx1ABCnop[[ix]], {ix, nx}];
invhosttandemitxx1 = Table[If[satx[[ix]], 0, radioinvhosttandemitxx[[ix]], {ix, nx}];
invhosttandemadslx1 = Table[If[satx[[ix]], 0, radioinvhosttandemadslx[[ix]], {ix, nx}];
invhosttandemTOTALx1 = Table[If[satx[[ix]], 0, radioinvhosttandemTOTALx[[ix]], {ix, nx}];
(* Compila o resume el total de inversiones a nivel de equipos a nivel HT imputable a todos los
servicios. Lo que manda es la tecnología de Tx de la Tandem. Inicialmente, si la Tandem no es satelital,
el nivel de inversión válido para cada una de las centrales del departamento será radioinvhosttandemTOTALx,
radioinvhosttandemx, radioinvhosttandemalqx1A1, radioinvhosttandemalqx1B1, radioinvhosttandemalqx1C1,
radioinvhosttandemalqx1LL, radioinvhosttandemalqx1LP, radioinvhosttandemalqx1AS, radioinvhosttandemalqx1BS,
radioinvhosttandemalqx1CS, radioinvhosttandemalqx1ABC2, radioinvhosttandemalqx1ABCnop,
radioinvhosttandemitxx, radioinvhosttandemadslx. Dependiendo de cada caso*)
Do[If[fiberx[[ix]],
  invhosttandemTOTALx1[[ix]] = fiberinvhosttandemTOTALx1[[ix]];
  invhosttandemx1[[ix]] = fiberinvhosttandemx1[[ix]];
  If[loadcompHTx[[ix]] > 0,
    invhosttandemLocx1[[ix]] = invhosttandemx1[[ix]]  $\frac{\text{loadcompHTLocx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ;
    invhosttandemLDNx1[[ix]] = invhosttandemx1[[ix]]  $\frac{\text{loadcompHTLDNx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ;
    invhosttandemLDIx1[[ix]] = invhosttandemx1[[ix]]  $\frac{\text{loadcompHTLDIx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ];
  (* Compila o resume el total de inversiones a nivel
  de equipos a nivel HT imputable a los servicios Local, LDN y LDI de Voz. *)
  invhosttandemalqx1A1[[ix]] = fiberinvhosttandemalqx1A1[[ix]];
  invhosttandemalqx1B1[[ix]] = fiberinvhosttandemalqx1B1[[ix]];
  invhosttandemalqx1C1[[ix]] = fiberinvhosttandemalqx1C1[[ix]];
  invhosttandemalqx1LL[[ix]] = fiberinvhosttandemalqx1LL[[ix]];
  invhosttandemalqx1LP[[ix]] = fiberinvhosttandemalqx1LP[[ix]];
  invhosttandemalqx1AS[[ix]] = fiberinvhosttandemalqx1AS[[ix]];
  invhosttandemalqx1BS[[ix]] = fiberinvhosttandemalqx1BS[[ix]];
  invhosttandemalqx1CS[[ix]] = fiberinvhosttandemalqx1CS[[ix]];
  invhosttandemalqx1ABC2[[ix]] = fiberinvhosttandemalqx1ABC2[[ix]];
  invhosttandemalqx1ABCnop[[ix]] = fiberinvhosttandemalqx1ABCnop[[ix]];
  invhosttandemitxx1[[ix]] = fiberinvhosttandemitxx1[[ix]];
  invhosttandemadslx1[[ix]] = fiberinvhosttandemadslx1[[ix]];
  (* Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las
  centrales del departamento será en cada caso: fiberinvhosttandemTOTALx1, fiberinvhosttandemx1,
  fiberinvhosttandemalqx1A1, fiberinvhosttandemalqx1B1, fiberinvhosttandemalqx1C1, fiberinvhosttandemalqx1LL,
  fiberinvhosttandemalqx1LP, fiberinvhosttandemalqx1AS, fiberinvhosttandemalqx1BS,
  fiberinvhosttandemalqx1CS, fiberinvhosttandemalqx1ABC2, fiberinvhosttandemalqx1ABCnop,
  fiberinvhosttandemitxx1, fiberinvhosttandemadslx1 dependiendo de cada servicio *)
  invhosttandemTOTALx1[[ix]] = invhosttandemTOTALx1[[ix]] + CosteAñadidoTrxPorDistanciaHTTOTALx[[ix]];
  invhosttandemx1[[ix]] = invhosttandemx1[[ix]] + CosteAñadidoTrxPorDistanciaHTx[[ix]];
  invhosttandemalqx1A1[[ix]] = invhosttandemalqx1A1[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1A1[[ix]];
  invhosttandemalqx1B1[[ix]] = invhosttandemalqx1B1[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1B1[[ix]];
  invhosttandemalqx1C1[[ix]] = invhosttandemalqx1C1[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1C1[[ix]];
  invhosttandemalqx1LL[[ix]] = invhosttandemalqx1LL[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1LL[[ix]];
  invhosttandemalqx1LP[[ix]] = invhosttandemalqx1LP[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1LP[[ix]];
  invhosttandemalqx1AS[[ix]] = invhosttandemalqx1AS[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1AS[[ix]];
  invhosttandemalqx1BS[[ix]] = invhosttandemalqx1BS[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1BS[[ix]];
  invhosttandemalqx1CS[[ix]] = invhosttandemalqx1CS[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1CS[[ix]];
  invhosttandemalqx1ABC2[[ix]] = invhosttandemalqx1ABC2[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1ABC2[[ix]];
  invhosttandemalqx1ABCnop[[ix]] = invhosttandemalqx1ABCnop[[ix]] + CosteAñadidoTrxPorDistanciaaalqHTx1ABCnop[[ix]];
  invhosttandemitxx1[[ix]] = invhosttandemitxx1[[ix]] + CosteAñadidoTrxPorDistanciaaitxHTx[[ix]];
  invhosttandemadslx1[[ix]] = invhosttandemadslx1[[ix]] + CosteAñadidoTrxPorDistanciaadslHTx[[ix]];
  , {ix, nx}];
(* Luego, identificado el nivel de inversión a tomar en cuenta en cada central,
se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], invhosttandemx1[[ix]], invhosttandemLocx1[[ix]], invhosttandemLDNx1[[ix]],
  invhosttandemLDIx1[[ix]], invhosttandemalqx1A1[[ix]], invhosttandemalqx1B1[[ix]], invhosttandemalqx1C1[[ix]],
  invhosttandemalqx1LL[[ix]], invhosttandemalqx1LP[[ix]], invhosttandemalqx1AS[[ix]], invhosttandemalqx1BS[[ix]],
  invhosttandemalqx1CS[[ix]], invhosttandemalqx1ABC2[[ix]], invhosttandemalqx1ABCnop[[ix]], invhosttandemadslx1[[ix]],
  invhosttandemitxx1[[ix]], invhosttandemTOTALx1[[ix]]}, {ix, nx}], TableDirections -> {Column, Row},

```



```

TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "idx", "Nombre Central", "invhosttandemx1",
"invhosttandemLocx1", "invhosttandemLDNx1", "invhosttandemLDIx1", "invhosttandemalqx1A1",
"invhosttandemalqx1B1", "invhosttandemalqx1C1", "invhosttandemalqx1LL", "invhosttandemalqx1LP",
"invhosttandemalqx1AS", "invhosttandemalqx1BS", "invhosttandemalqx1CS", "invhosttandemalqx1ABC2",
"invhosttandemalqx1ABCnop", "invhosttandemadslx1", "invhosttandemitxx1", "invhosttandemTOTALx1"}}];

invhosttandemd1 = zerod;
invhosttandemLocd1 = zerod;
invhosttandemLDNd1 = zerod;
invhosttandemLDId1 = zerod;
invhosttandemalqd1A1 = zerod;
invhosttandemalqd1B1 = zerod;
invhosttandemalqd1C1 = zerod;
invhosttandemalqd1LL = zerod;
invhosttandemalqd1LP = zerod;
invhosttandemalqd1AS = zerod;
invhosttandemalqd1BS = zerod;
invhosttandemalqd1CS = zerod;
invhosttandemalqd1ABC2 = zerod;
invhosttandemalqd1ABCnop = zerod;
invhosttandemitxd1 = zerod;
invhosttandemadsl1 = zerod;
invhosttandemTOTALd1 = zerod;
Do[
  invhosttandemd1[[deptx[[ix]]]] = invhosttandemd1[[deptx[[ix]]]] + invhosttandemx1[[ix]];
  invhosttandemLocd1[[deptx[[ix]]]] = invhosttandemLocd1[[deptx[[ix]]]] + invhosttandemLocx1[[ix]];
  invhosttandemLDNd1[[deptx[[ix]]]] = invhosttandemLDNd1[[deptx[[ix]]]] + invhosttandemLDNx1[[ix]];
  invhosttandemLDId1[[deptx[[ix]]]] = invhosttandemLDId1[[deptx[[ix]]]] + invhosttandemLDIx1[[ix]];
  invhosttandemalqd1A1[[deptx[[ix]]]] = invhosttandemalqd1A1[[deptx[[ix]]]] + invhosttandemalqx1A1[[ix]];
  invhosttandemalqd1B1[[deptx[[ix]]]] = invhosttandemalqd1B1[[deptx[[ix]]]] + invhosttandemalqx1B1[[ix]];
  invhosttandemalqd1C1[[deptx[[ix]]]] = invhosttandemalqd1C1[[deptx[[ix]]]] + invhosttandemalqx1C1[[ix]];
  invhosttandemalqd1LL[[deptx[[ix]]]] = invhosttandemalqd1LL[[deptx[[ix]]]] + invhosttandemalqx1LL[[ix]];
  invhosttandemalqd1LP[[deptx[[ix]]]] = invhosttandemalqd1LP[[deptx[[ix]]]] + invhosttandemalqx1LP[[ix]];
  invhosttandemalqd1AS[[deptx[[ix]]]] = invhosttandemalqd1AS[[deptx[[ix]]]] + invhosttandemalqx1AS[[ix]];
  invhosttandemalqd1BS[[deptx[[ix]]]] = invhosttandemalqd1BS[[deptx[[ix]]]] + invhosttandemalqx1BS[[ix]];
  invhosttandemalqd1CS[[deptx[[ix]]]] = invhosttandemalqd1CS[[deptx[[ix]]]] + invhosttandemalqx1CS[[ix]];
  invhosttandemalqd1ABC2[[deptx[[ix]]]] = invhosttandemalqd1ABC2[[deptx[[ix]]]] + invhosttandemalqx1ABC2[[ix]];
  invhosttandemalqd1ABCnop[[deptx[[ix]]]] = invhosttandemalqd1ABCnop[[deptx[[ix]]]] + invhosttandemalqx1ABCnop[[ix]];
  invhosttandemitxd1[[deptx[[ix]]]] = invhosttandemitxd1[[deptx[[ix]]]] + invhosttandemitxx1[[ix]];
  invhosttandemadsl1[[deptx[[ix]]]] = invhosttandemadsl1[[deptx[[ix]]]] + invhosttandemadslx1[[ix]];
  invhosttandemTOTALd1[[deptx[[ix]]]] = invhosttandemTOTALd1[[deptx[[ix]]]] + invhosttandemTOTALx1[[ix]];
  , {ix, nx}];
(* Acumula a nivel departamental las variables invhosttandemx1, invhosttandemLocx1, invhosttandemLDNx1,
invhosttandemLDIx1, invhosttandemalqx1A1, invhosttandemalqx1B1, invhosttandemalqx1C1,
invhosttandemalqx1LL, invhosttandemalqx1LP, invhosttandemalqx1AS, invhosttandemalqx1BS,
invhosttandemalqx1CS, invhosttandemalqx1ABC2, invhosttandemalqx1ABCnop, invhosttandemitxx1,
invhosttandemadslx1 y invhosttandemTOTALx1 imputable a cada uno de los servicios *)

Print[TableForm[Table[{id, deptTXTd[[id]], invhosttandemd1[[id]], invhosttandemLocd1[[id]], invhosttandemLDNd1[[id]],
  invhosttandemLDId1[[id]], invhosttandemalqd1A1[[id]], invhosttandemalqd1B1[[id]], invhosttandemalqd1C1[[id]],
  invhosttandemalqd1LL[[id]], invhosttandemalqd1LP[[id]], invhosttandemalqd1AS[[id]], invhosttandemalqd1BS[[id]],
  invhosttandemalqd1CS[[id]], invhosttandemalqd1ABC2[[id]], invhosttandemalqd1ABCnop[[id]], invhosttandemadsl1[[id]],
  invhosttandemitxd1[[id]], invhosttandemTOTALd1[[id]]}, {id, nd}], TableDirections → {Column, Row},
TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "invhosttandemd1",
"invhosttandemLocd1", "invhosttandemLDNd1", "invhosttandemLDId1", "invhosttandemalqd1A1",
"invhosttandemalqd1B1", "invhosttandemalqd1C1", "invhosttandemalqd1LL", "invhosttandemalqd1LP",
"invhosttandemalqd1AS", "invhosttandemalqd1BS", "invhosttandemalqd1CS", "invhosttandemalqd1ABC2",
"invhosttandemalqd1ABCnop", "invhosttandemadsl1", "invhosttandemitxd1", "invhosttandemTOTALd1"}}];

invhosttandemx2 = zerox;
invhosttandemLocx2 = zerox;
invhosttandemLDNx2 = zerox;
invhosttandemLDIx2 = zerox;
invhosttandemalqx2A1 = zerox;
invhosttandemalqx2B1 = zerox;
invhosttandemalqx2C1 = zerox;
invhosttandemalqx2LL = zerox;
invhosttandemalqx2LP = zerox;
invhosttandemalqx2AS = zerox;
invhosttandemalqx2BS = zerox;
invhosttandemalqx2CS = zerox;
invhosttandemalqx2ABC2 = zerox;
invhosttandemalqx2ABCnop = zerox;
invhosttandemitxx2 = zerox;
invhosttandemadslx2 = zerox;
invhosttandemTOTALx2 = zerox;
Do[If[fiberx[[ix]],
  invhosttandemx2[[ix]] = fiberinvhosttandemx2[[ix]];

```

```
If [loadcompHTx[[ix]] > 0,
```

```
    invhosttandemLocx2[[ix]] = invhosttandemx2[[ix]]  $\frac{\text{loadcompHTLocx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ;
```

```
    invhosttandemLDNx2[[ix]] = invhosttandemx2[[ix]]  $\frac{\text{loadcompHTLDNx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ;
```

```
    invhosttandemLDIx2[[ix]] = invhosttandemx2[[ix]]  $\frac{\text{loadcompHTLDIx}[[ix]]}{\text{loadcompHTx}[[ix]]}$ ];];
```

(* Compila o resume el total de inversiones aplicable a los cables imputable a los servicios Local, LDN y LDI de Voz. Como dicha información sólo es aplicable para el uso de fibra, en todos los demás casos es cero *)

```
invhosttandemalqx2A1[[ix]] = fiberinvhosttandemalqx2A1[[ix]];
```

```
invhosttandemalqx2B1[[ix]] = fiberinvhosttandemalqx2B1[[ix]];
```

```
invhosttandemalqx2C1[[ix]] = fiberinvhosttandemalqx2C1[[ix]];
```

```
invhosttandemalqx2LL[[ix]] = fiberinvhosttandemalqx2LL[[ix]];
```

```
invhosttandemalqx2LP[[ix]] = fiberinvhosttandemalqx2LP[[ix]];
```

```
invhosttandemalqx2AS[[ix]] = fiberinvhosttandemalqx2AS[[ix]];
```

```
invhosttandemalqx2BS[[ix]] = fiberinvhosttandemalqx2BS[[ix]];
```

```
invhosttandemalqx2CS[[ix]] = fiberinvhosttandemalqx2CS[[ix]];
```

```
invhosttandemalqx2ABC2[[ix]] = fiberinvhosttandemalqx2ABC2[[ix]];
```

```
invhosttandemalqx2ABCnop[[ix]] = fiberinvhosttandemalqx2ABCnop[[ix]];
```

```
invhosttandemitxx2[[ix]] = fiberinvhosttandemitxx2[[ix]];
```

```
invhosttandemadslx2[[ix]] = fiberinvhosttandemadslx2[[ix]];
```

```
invhosttandemTOTALx2[[ix]] = fiberinvhosttandemTOTALx2[[ix]]];];
```

```
, {ix, nx}];
```

(* Compila o resume el total de inversiones aplicable a los cables imputable a cada servicio. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvhosttandemx2, fiberinvhosttandemalqx2A1, fiberinvhosttandemalqx2B1, fiberinvhosttandemalqx2C1, fiberinvhosttandemalqx2LL, fiberinvhosttandemalqx2LP, fiberinvhosttandemalqx2AS, fiberinvhosttandemalqx2BS, fiberinvhosttandemalqx2CS, fiberinvhosttandemalqx2ABC2, fiberinvhosttandemalqx2ABCnop, fiberinvhosttandemitxx2, fiberinvhosttandemadslx2 o fiberinvhosttandemTOTALx2 según sea el caso, en todos los demás casos es cero *)

```
Print[TableForm[Table[{ix, idx[[ix]], y[[ix, 13]], invhosttandemx2[[ix]], invhosttandemLocx2[[ix]], invhosttandemLDNx2[[ix]],
    invhosttandemLDIx2[[ix]], invhosttandemalqx2A1[[ix]], invhosttandemalqx2B1[[ix]], invhosttandemalqx2C1[[ix]],
    invhosttandemalqx2LL[[ix]], invhosttandemalqx2LP[[ix]], invhosttandemalqx2AS[[ix]], invhosttandemalqx2BS[[ix]],
    invhosttandemalqx2CS[[ix]], invhosttandemalqx2ABC2[[ix]], invhosttandemalqx2ABCnop[[ix]], invhosttandemadslx2[[ix]],
    invhosttandemitxx2[[ix]], invhosttandemTOTALx2[[ix]]}, {ix, nx}], TableDirections → {Column, Row},
    TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "idx", "Nombre Central", "invhosttandemx2",
    "invhosttandemLocx2", "invhosttandemLDNx2", "invhosttandemLDIx2", "invhosttandemalqx2A1",
    "invhosttandemalqx2B1", "invhosttandemalqx2C1", "invhosttandemalqx2LL", "invhosttandemalqx2LP",
    "invhosttandemalqx2AS", "invhosttandemalqx2BS", "invhosttandemalqx2CS", "invhosttandemalqx2ABC2",
    "invhosttandemalqx2ABCnop", "invhosttandemadslx2", "invhosttandemitxx2", "invhosttandemTOTALx2"}}]]];
```

```
invhosttandemd2 = zerod;
```

```
invhosttandemLocd2 = zerod;
```

```
invhosttandemLDNd2 = zerod;
```

```
invhosttandemLDId2 = zerod;
```

```
invhosttandemalqd2A1 = zerod;
```

```
invhosttandemalqd2B1 = zerod;
```

```
invhosttandemalqd2C1 = zerod;
```

```
invhosttandemalqd2LL = zerod;
```

```
invhosttandemalqd2LP = zerod;
```

```
invhosttandemalqd2AS = zerod;
```

```
invhosttandemalqd2BS = zerod;
```

```
invhosttandemalqd2CS = zerod;
```

```
invhosttandemalqd2ABC2 = zerod;
```

```
invhosttandemalqd2ABCnop = zerod;
```

```
invhosttandemitxd2 = zerod;
```

```
invhosttandemadslid2 = zerod;
```

```
invhosttandemTOTALd2 = zerod;
```

```
Do[
```

```
    invhosttandemd2[deptx[[ix]]] = invhosttandemd2[deptx[[ix]]] + invhosttandemx2[[ix]];
```

```
    invhosttandemLocd2[deptx[[ix]]] = invhosttandemLocd2[deptx[[ix]]] + invhosttandemLocx2[[ix]];
```

```
    invhosttandemLDNd2[deptx[[ix]]] = invhosttandemLDNd2[deptx[[ix]]] + invhosttandemLDNx2[[ix]];
```

```
    invhosttandemLDId2[deptx[[ix]]] = invhosttandemLDId2[deptx[[ix]]] + invhosttandemLDIx2[[ix]];
```

```
    invhosttandemalqd2A1[deptx[[ix]]] = invhosttandemalqd2A1[deptx[[ix]]] + invhosttandemalqx2A1[[ix]];
```

```
    invhosttandemalqd2B1[deptx[[ix]]] = invhosttandemalqd2B1[deptx[[ix]]] + invhosttandemalqx2B1[[ix]];
```

```
    invhosttandemalqd2C1[deptx[[ix]]] = invhosttandemalqd2C1[deptx[[ix]]] + invhosttandemalqx2C1[[ix]];
```

```
    invhosttandemalqd2LL[deptx[[ix]]] = invhosttandemalqd2LL[deptx[[ix]]] + invhosttandemalqx2LL[[ix]];
```

```
    invhosttandemalqd2LP[deptx[[ix]]] = invhosttandemalqd2LP[deptx[[ix]]] + invhosttandemalqx2LP[[ix]];
```

```
    invhosttandemalqd2AS[deptx[[ix]]] = invhosttandemalqd2AS[deptx[[ix]]] + invhosttandemalqx2AS[[ix]];
```

```
    invhosttandemalqd2BS[deptx[[ix]]] = invhosttandemalqd2BS[deptx[[ix]]] + invhosttandemalqx2BS[[ix]];
```

```
    invhosttandemalqd2CS[deptx[[ix]]] = invhosttandemalqd2CS[deptx[[ix]]] + invhosttandemalqx2CS[[ix]];
```

```
    invhosttandemalqd2ABC2[deptx[[ix]]] = invhosttandemalqd2ABC2[deptx[[ix]]] + invhosttandemalqx2ABC2[[ix]];
```

```
    invhosttandemalqd2ABCnop[deptx[[ix]]] = invhosttandemalqd2ABCnop[deptx[[ix]]] + invhosttandemalqx2ABCnop[[ix]];
```

```
    invhosttandemitxd2[deptx[[ix]]] = invhosttandemitxd2[deptx[[ix]]] + invhosttandemitxx2[[ix]];
```

```
    invhosttandemadslid2[deptx[[ix]]] = invhosttandemadslid2[deptx[[ix]]] + invhosttandemadslx2[[ix]];
```



```

invhosttandemTOTALd2[deptx[ix]] = invhosttandemTOTALd2[deptx[ix]] + invhosttandemTOTALx2[ix];
, {ix, nx}];

(* Acumula a nivel departamental las variables invhosttandemx2, invhosttandemLocx2, invhosttandemLDNx2,
invhosttandemLDIx2, invhosttandemalqx2A1, invhosttandemalqx2B1, invhosttandemalqx2C1,
invhosttandemalqx2LL, invhosttandemalqx2LP, invhosttandemalqx2AS, invhosttandemalqx2BS,
invhosttandemalqx2CS, invhosttandemalqx2ABC2, invhosttandemalqx2ABCnop, invhosttandemitxx2,
invhosttandemadslx2 y invhosttandemTOTALx2 imputable a cada uno de los servicios *)

Print[TableForm[Table[{id, deptTXTd[id], invhosttandemd2[id], invhosttandemLocd2[id], invhosttandemLDNd2[id],
invhosttandemLDId2[id], invhosttandemalqd2A1[id], invhosttandemalqd2B1[id], invhosttandemalqd2C1[id],
invhosttandemalqd2LL[id], invhosttandemalqd2LP[id], invhosttandemalqd2AS[id], invhosttandemalqd2BS[id],
invhosttandemalqd2CS[id], invhosttandemalqd2ABC2[id], invhosttandemalqd2ABCnop[id], invhosttandemadsl2[id],
invhosttandemitxd2[id], invhosttandemTOTALd2[id]}, {id, nd}], TableDirections → {Column, Row},
TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "invhosttandemd2",
"invhosttandemLocd2", "invhosttandemLDNd2", "invhosttandemLDId2", "invhosttandemalqd2A1",
"invhosttandemalqd2B1", "invhosttandemalqd2C1", "invhosttandemalqd2LL", "invhosttandemalqd2LP",
"invhosttandemalqd2AS", "invhosttandemalqd2BS", "invhosttandemalqd2CS", "invhosttandemalqd2ABC2",
"invhosttandemalqd2ABCnop", "invhosttandemadsl2", "invhosttandemitxd2", "invhosttandemTOTALd2"}}]];

invhosttandemx3 = zerox;
invhosttandemLocx3 = zerox;
invhosttandemLDNx3 = zerox;
invhosttandemLDIx3 = zerox;
invhosttandemalqx3A1 = zerox;
invhosttandemalqx3B1 = zerox;
invhosttandemalqx3C1 = zerox;
invhosttandemalqx3LL = zerox;
invhosttandemalqx3LP = zerox;
invhosttandemalqx3AS = zerox;
invhosttandemalqx3BS = zerox;
invhosttandemalqx3CS = zerox;
invhosttandemalqx3ABC2 = zerox;
invhosttandemalqx3ABCnop = zerox;
invhosttandemitxx3 = zerox;
invhosttandemadslx3 = zerox;
invhosttandemTOTALx3 = zerox;

Do[If[fiberx[ix],
invhosttandemx3[ix] = fiberinvhosttandemx3[ix];
If[loadcompHTx[ix] > 0,

invhosttandemLocx3[ix] = invhosttandemx3[ix]  $\frac{\text{loadcompHTLocx}[\text{ix}]}{\text{loadcompHTx}[\text{ix}]}$ ;

invhosttandemLDNx3[ix] = invhosttandemx3[ix]  $\frac{\text{loadcompHTLDNx}[\text{ix}]}{\text{loadcompHTx}[\text{ix}]}$ ;

invhosttandemLDIx3[ix] = invhosttandemx3[ix]  $\frac{\text{loadcompHTLDIx}[\text{ix}]}{\text{loadcompHTx}[\text{ix}]}$ ];

(* Compila o resume el total de inversiones aplicable a obras civiles imputable a los servicios Local,
LDN y LDI de Voz. Como dicha información sólo es aplicable para el uso de fibra,
en todos los demás casos es cero *)
invhosttandemalqx3A1[ix] = fiberinvhosttandemalqx3A1[ix];
invhosttandemalqx3B1[ix] = fiberinvhosttandemalqx3B1[ix];
invhosttandemalqx3C1[ix] = fiberinvhosttandemalqx3C1[ix];
invhosttandemalqx3LL[ix] = fiberinvhosttandemalqx3LL[ix];
invhosttandemalqx3LP[ix] = fiberinvhosttandemalqx3LP[ix];
invhosttandemalqx3AS[ix] = fiberinvhosttandemalqx3AS[ix];
invhosttandemalqx3BS[ix] = fiberinvhosttandemalqx3BS[ix];
invhosttandemalqx3CS[ix] = fiberinvhosttandemalqx3CS[ix];
invhosttandemalqx3ABC2[ix] = fiberinvhosttandemalqx3ABC2[ix];
invhosttandemalqx3ABCnop[ix] = fiberinvhosttandemalqx3ABCnop[ix];
invhosttandemitxx3[ix] = fiberinvhosttandemitxx3[ix];
invhosttandemadslx3[ix] = fiberinvhosttandemadslx3[ix];
invhosttandemTOTALx3[ix] = fiberinvhosttandemTOTALx3[ix];

, {ix, nx}];

(* Compila o resume el total de inversiones aplicable a obras civiles imputable
a cada servicio. Como dicha información sólo es aplicable para el uso de fibra,
simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvhosttandemx3,
fiberinvhosttandemalqx3A1, fiberinvhosttandemalqx3B1, fiberinvhosttandemalqx3C1,
fiberinvhosttandemalqx3LL, fiberinvhosttandemalqx3LP, fiberinvhosttandemalqx3AS,
fiberinvhosttandemalqx3BS, fiberinvhosttandemalqx3CS, fiberinvhosttandemalqx3ABC2,
fiberinvhosttandemalqx3ABCnop, fiberinvhosttandemitxx3, fiberinvhosttandemadslx3,
fiberinvhosttandemTOTALx3, según sea el caso, en todos los demás casos es cero *)

Print[TableForm[Table[{ix, idx[ix], y[ix, 13], invhosttandemx3[ix], invhosttandemLocx3[ix], invhosttandemLDNx3[ix],
invhosttandemLDIx3[ix], invhosttandemalqx3A1[ix], invhosttandemalqx3B1[ix], invhosttandemalqx3C1[ix],
invhosttandemalqx3LL[ix], invhosttandemalqx3LP[ix], invhosttandemalqx3AS[ix], invhosttandemalqx3BS[ix],
invhosttandemalqx3CS[ix], invhosttandemalqx3ABC2[ix], invhosttandemalqx3ABCnop[ix], invhosttandemadslx3[ix],
invhosttandemitxx3[ix], invhosttandemTOTALx3[ix]}, {ix, nx}], TableDirections → {Column, Row},
TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "idx", "Nombre Central", "invhosttandemx3",

```

```
"invhosttandemLocx3", "invhosttandemLDNx3", "invhosttandemLDIx3", "invhosttandemalqx3A1",
"invhosttandemalqx3B1", "invhosttandemalqx3C1", "invhosttandemalqx3LL", "invhosttandemalqx3LP",
"invhosttandemalqx3AS", "invhosttandemalqx3BS", "invhosttandemalqx3CS", "invhosttandemalqx3ABC2",
"invhosttandemalqx3ABCnop", "invhosttandemadslx3", "invhosttandemitxx3", "invhosttandemTOTALx3"}]]];
```

```
invhosttandemd3 = zerod;
invhosttandemLocd3 = zerod;
invhosttandemLDNd3 = zerod;
invhosttandemLDId3 = zerod;
invhosttandemalqd3A1 = zerod;
invhosttandemalqd3B1 = zerod;
invhosttandemalqd3C1 = zerod;
invhosttandemalqd3LL = zerod;
invhosttandemalqd3LP = zerod;
invhosttandemalqd3AS = zerod;
invhosttandemalqd3BS = zerod;
invhosttandemalqd3CS = zerod;
invhosttandemalqd3ABC2 = zerod;
invhosttandemalqd3ABCnop = zerod;
invhosttandemitxd3 = zerod;
invhosttandemadsl3 = zerod;
invhosttandemTOTALd3 = zerod;
```

```
Do[
  invhosttandemd3[[deptx[ix]]] = invhosttandemd3[[deptx[ix]]] + invhosttandemx3[[ix]];
  invhosttandemLocd3[[deptx[ix]]] = invhosttandemLocd3[[deptx[ix]]] + invhosttandemLocx3[[ix]];
  invhosttandemLDNd3[[deptx[ix]]] = invhosttandemLDNd3[[deptx[ix]]] + invhosttandemLDNx3[[ix]];
  invhosttandemLDId3[[deptx[ix]]] = invhosttandemLDId3[[deptx[ix]]] + invhosttandemLDIx3[[ix]];
  invhosttandemalqd3A1[[deptx[ix]]] = invhosttandemalqd3A1[[deptx[ix]]] + invhosttandemalqx3A1[[ix]];
  invhosttandemalqd3B1[[deptx[ix]]] = invhosttandemalqd3B1[[deptx[ix]]] + invhosttandemalqx3B1[[ix]];
  invhosttandemalqd3C1[[deptx[ix]]] = invhosttandemalqd3C1[[deptx[ix]]] + invhosttandemalqx3C1[[ix]];
  invhosttandemalqd3LL[[deptx[ix]]] = invhosttandemalqd3LL[[deptx[ix]]] + invhosttandemalqx3LL[[ix]];
  invhosttandemalqd3LP[[deptx[ix]]] = invhosttandemalqd3LP[[deptx[ix]]] + invhosttandemalqx3LP[[ix]];
  invhosttandemalqd3AS[[deptx[ix]]] = invhosttandemalqd3AS[[deptx[ix]]] + invhosttandemalqx3AS[[ix]];
  invhosttandemalqd3BS[[deptx[ix]]] = invhosttandemalqd3BS[[deptx[ix]]] + invhosttandemalqx3BS[[ix]];
  invhosttandemalqd3CS[[deptx[ix]]] = invhosttandemalqd3CS[[deptx[ix]]] + invhosttandemalqx3CS[[ix]];
  invhosttandemalqd3ABC2[[deptx[ix]]] = invhosttandemalqd3ABC2[[deptx[ix]]] + invhosttandemalqx3ABC2[[ix]];
  invhosttandemalqd3ABCnop[[deptx[ix]]] = invhosttandemalqd3ABCnop[[deptx[ix]]] + invhosttandemalqx3ABCnop[[ix]];
  invhosttandemitxd3[[deptx[ix]]] = invhosttandemitxd3[[deptx[ix]]] + invhosttandemitxx3[[ix]];
  invhosttandemadsl3[[deptx[ix]]] = invhosttandemadsl3[[deptx[ix]]] + invhosttandemadslx3[[ix]];
  invhosttandemTOTALd3[[deptx[ix]]] = invhosttandemTOTALd3[[deptx[ix]]] + invhosttandemTOTALx3[[ix]];
, {ix, nx}];
```

```
(* Acumula a nivel departamental las variables invhosttandemx3, invhosttandemLocx3, invhosttandemLDNx3,
invhosttandemLDIx3, invhosttandemalqx3A1, invhosttandemalqx3B1, invhosttandemalqx3C1,
invhosttandemalqx3LL, invhosttandemalqx3LP, invhosttandemalqx3AS, invhosttandemalqx3BS,
invhosttandemalqx3CS, invhosttandemalqx3ABC2, invhosttandemalqx3ABCnop, invhosttandemitxx3,
invhosttandemadslx3 y invhosttandemTOTALx3 imputable a cada uno de los servicios *)
```

```
Print[TableForm[Table[{id, deptTXTd[id], invhosttandemd3[id], invhosttandemLocd3[id], invhosttandemLDNd3[id],
  invhosttandemLDId3[id], invhosttandemalqd3A1[id], invhosttandemalqd3B1[id], invhosttandemalqd3C1[id],
  invhosttandemalqd3LL[id], invhosttandemalqd3LP[id], invhosttandemalqd3AS[id], invhosttandemalqd3BS[id],
  invhosttandemalqd3CS[id], invhosttandemalqd3ABC2[id], invhosttandemalqd3ABCnop[id], invhosttandemadsl3[id],
  invhosttandemitxd3[id], invhosttandemTOTALd3[id]}, {id, nd}], TableDirections → {Column, Row},
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "invhosttandemd3",
  "invhosttandemLocd3", "invhosttandemLDNd3", "invhosttandemLDId3", "invhosttandemalqd3A1",
  "invhosttandemalqd3B1", "invhosttandemalqd3C1", "invhosttandemalqd3LL", "invhosttandemalqd3LP",
  "invhosttandemalqd3AS", "invhosttandemalqd3BS", "invhosttandemalqd3CS", "invhosttandemalqd3ABC2",
  "invhosttandemalqd3ABCnop", "invhosttandemadsl3", "invhosttandemitxd3", "invhosttandemTOTALd3"}]]];
```

```
invhosttandemx = invhosttandemx1 + invhosttandemx2 + invhosttandemx3;
invhosttandemLocx = invhosttandemLocx1 + invhosttandemLocx2 + invhosttandemLocx3;
invhosttandemLDNx = invhosttandemLDNx1 + invhosttandemLDNx2 + invhosttandemLDNx3;
invhosttandemLDIx = invhosttandemLDIx1 + invhosttandemLDIx2 + invhosttandemLDIx3;
invhosttandemalqx1 = invhosttandemalqx1A1 + invhosttandemalqx2A1 + invhosttandemalqx3A1;
invhosttandemalqx1B1 = invhosttandemalqx1B1 + invhosttandemalqx2B1 + invhosttandemalqx3B1;
invhosttandemalqx1C1 = invhosttandemalqx1C1 + invhosttandemalqx2C1 + invhosttandemalqx3C1;
invhosttandemalqx1LL = invhosttandemalqx1LL + invhosttandemalqx2LL + invhosttandemalqx3LL;
invhosttandemalqx1LP = invhosttandemalqx1LP + invhosttandemalqx2LP + invhosttandemalqx3LP;
invhosttandemalqx1AS = invhosttandemalqx1AS + invhosttandemalqx2AS + invhosttandemalqx3AS;
invhosttandemalqx1BS = invhosttandemalqx1BS + invhosttandemalqx2BS + invhosttandemalqx3BS;
invhosttandemalqx1CS = invhosttandemalqx1CS + invhosttandemalqx2CS + invhosttandemalqx3CS;
invhosttandemalqx1ABC2 = invhosttandemalqx1ABC2 + invhosttandemalqx2ABC2 + invhosttandemalqx3ABC2;
invhosttandemalqx1ABCnop = invhosttandemalqx1ABCnop + invhosttandemalqx2ABCnop + invhosttandemalqx3ABCnop;
invhosttandemitxx = invhosttandemitxx1 + invhosttandemitxx2 + invhosttandemitxx3;
invhosttandemadslx = invhosttandemadslx1 + invhosttandemadslx2 + invhosttandemadslx3;
invhosttandemTOTALx = invhosttandemTOTALx1 + invhosttandemTOTALx2 + invhosttandemTOTALx3;
(* Total de inversiones en Tx a nivel HT imputable a cada servicio *)
```

```
Print[TableForm[Table[{ix, idx[ix], y[ix, 13], invhosttandemx[ix], invhosttandemLocx[ix], invhosttandemLDNx[ix],
  invhosttandemLDIx[ix], invhosttandemalqx1A1[ix], invhosttandemalqx1B1[ix], invhosttandemalqx1C1[ix],
```

```

invhosttandemalqxLL[[ix]], invhosttandemalqxLP[[ix]], invhosttandemalqxAS[[ix]], invhosttandemalqxBS[[ix]],
invhosttandemalqxCS[[ix]], invhosttandemalqxABC2[[ix]], invhosttandemalqxABCnop[[ix]], invhosttandemadslx[[ix]],
invhosttandemitxx[[ix]], invhosttandemTOTALx[[ix]]}, {ix, nx}}, TableDirections → {Column, Row},
TableSpacing → {0, 1}, TableHeadings → {None, {"ix", "idx", "Nombre Central", "invhosttandemx",
"invhosttandemLocx", "invhosttandemLDNx", "invhosttandemLDIx", "invhosttandemalqxA1",
"invhosttandemalqxB1", "invhosttandemalqxC1", "invhosttandemalqxLL", "invhosttandemalqxLP",
"invhosttandemalqxAS", "invhosttandemalqxBS", "invhosttandemalqxCS", "invhosttandemalqxABC2",
"invhosttandemalqxABCnop", "invhosttandemadslx", "invhosttandemitxx", "invhosttandemTOTALx"}}}];

```

```

invhosttandemd = zerod;
invhosttandemLocd = zerod;
invhosttandemLDNd = zerod;
invhosttandemLDId = zerod;
invhosttandemalqdA1 = zerod;
invhosttandemalqdB1 = zerod;
invhosttandemalqdc1 = zerod;
invhosttandemalqdLL = zerod;
invhosttandemalqdLP = zerod;
invhosttandemalqdAS = zerod;
invhosttandemalqdBs = zerod;
invhosttandemalqdCS = zerod;
invhosttandemalqdABC2 = zerod;
invhosttandemalqdABCnop = zerod;
invhosttandemitxd = zerod;
invhosttandemadsl = zerod;
invhosttandemTOTALd = zerod;

```

```
Do[
```

```

    invhosttandemd[[deptx[[ix]]]] = invhosttandemd[[deptx[[ix]]]] + invhosttandemx[[ix]];
    invhosttandemLocd[[deptx[[ix]]]] = invhosttandemLocd[[deptx[[ix]]]] + invhosttandemLocx[[ix]];
    invhosttandemLDNd[[deptx[[ix]]]] = invhosttandemLDNd[[deptx[[ix]]]] + invhosttandemLDNx[[ix]];
    invhosttandemLDId[[deptx[[ix]]]] = invhosttandemLDId[[deptx[[ix]]]] + invhosttandemLDIx[[ix]];
    invhosttandemalqdA1[[deptx[[ix]]]] = invhosttandemalqdA1[[deptx[[ix]]]] + invhosttandemalqxA1[[ix]];
    invhosttandemalqdB1[[deptx[[ix]]]] = invhosttandemalqdB1[[deptx[[ix]]]] + invhosttandemalqxB1[[ix]];
    invhosttandemalqdc1[[deptx[[ix]]]] = invhosttandemalqdc1[[deptx[[ix]]]] + invhosttandemalqxC1[[ix]];
    invhosttandemalqdLL[[deptx[[ix]]]] = invhosttandemalqdLL[[deptx[[ix]]]] + invhosttandemalqxLL[[ix]];
    invhosttandemalqdLP[[deptx[[ix]]]] = invhosttandemalqdLP[[deptx[[ix]]]] + invhosttandemalqxLP[[ix]];
    invhosttandemalqdAS[[deptx[[ix]]]] = invhosttandemalqdAS[[deptx[[ix]]]] + invhosttandemalqxAS[[ix]];
    invhosttandemalqdBs[[deptx[[ix]]]] = invhosttandemalqdBs[[deptx[[ix]]]] + invhosttandemalqxBS[[ix]];
    invhosttandemalqdCS[[deptx[[ix]]]] = invhosttandemalqdCS[[deptx[[ix]]]] + invhosttandemalqxCS[[ix]];
    invhosttandemalqdABC2[[deptx[[ix]]]] = invhosttandemalqdABC2[[deptx[[ix]]]] + invhosttandemalqxABC2[[ix]];
    invhosttandemalqdABCnop[[deptx[[ix]]]] = invhosttandemalqdABCnop[[deptx[[ix]]]] + invhosttandemalqxABCnop[[ix]];
    invhosttandemitxd[[deptx[[ix]]]] = invhosttandemitxd[[deptx[[ix]]]] + invhosttandemitxx[[ix]];
    invhosttandemadsl[[deptx[[ix]]]] = invhosttandemadsl[[deptx[[ix]]]] + invhosttandemadslx[[ix]];
    invhosttandemTOTALd[[deptx[[ix]]]] = invhosttandemTOTALd[[deptx[[ix]]]] + invhosttandemTOTALx[[ix]];
    , {ix, nx}}];

```

```

(* Acumula a nivel departamental las variables invhosttandemx, invhosttandemLocx,
invhosttandemLDNx, invhosttandemLDIx, invhosttandemalqxA1, invhosttandemalqxB1, invhosttandemalqxC1,
invhosttandemalqxLL, invhosttandemalqxLP, invhosttandemalqxAS, invhosttandemalqxBs,
invhosttandemalqxCS, invhosttandemalqxABC2, invhosttandemalqxABCnop, invhosttandemitxx,
invhosttandemadslx y invhosttandemTOTALx imputable a cada uno de los servicios *)

```

```

Print[TableForm[Table[{id, deptTXTd[id]], invhosttandemd[id], invhosttandemLocd[id], invhosttandemLDNd[id],
    invhosttandemLDId[id], invhosttandemalqdA1[id], invhosttandemalqdB1[id], invhosttandemalqdc1[id],
    invhosttandemalqdLL[id], invhosttandemalqdLP[id], invhosttandemalqdAS[id], invhosttandemalqdBs[id],
    invhosttandemalqdCS[id], invhosttandemalqdABC2[id], invhosttandemalqdABCnop[id], invhosttandemadsl[id],
    invhosttandemitxd[id], invhosttandemTOTALd[id]}, {id, nd}], TableDirections → {Column, Row},
    TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "invhosttandemd",
    "invhosttandemLocd", "invhosttandemLDNd", "invhosttandemLDId", "invhosttandemalqdA1",
    "invhosttandemalqdB1", "invhosttandemalqdc1", "invhosttandemalqdLL", "invhosttandemalqdLP",
    "invhosttandemalqdAS", "invhosttandemalqdBs", "invhosttandemalqdCS", "invhosttandemalqdABC2",
    "invhosttandemalqdABCnop", "invhosttandemadsl", "invhosttandemitxd", "invhosttandemTOTALd"}}}];

```

```

invhosttandem = Plus @@ invhosttandemd;
invhosttandemLoc = Plus @@ invhosttandemLocd;
invhosttandemLDN = Plus @@ invhosttandemLDNd;
invhosttandemLDI = Plus @@ invhosttandemLDId;
invhosttandemalqA1 = Plus @@ invhosttandemalqdA1;
invhosttandemalqB1 = Plus @@ invhosttandemalqdB1;
invhosttandemalqC1 = Plus @@ invhosttandemalqdc1;
invhosttandemalqLL = Plus @@ invhosttandemalqdLL;
invhosttandemalqLP = Plus @@ invhosttandemalqdLP;
invhosttandemalqAS = Plus @@ invhosttandemalqdAS;
invhosttandemalqBs = Plus @@ invhosttandemalqdBs;
invhosttandemalqCS = Plus @@ invhosttandemalqdCS;
invhosttandemalqABC2 = Plus @@ invhosttandemalqdABC2;
invhosttandemalqABCnop = Plus @@ invhosttandemalqdABCnop;
invhosttandemitx = Plus @@ invhosttandemitxd;
invhosttandemadsl = Plus @@ invhosttandemadsl;
invhosttandemTOTAL = Plus @@ invhosttandemTOTALd;

```

```
(* Total de inversiones en Tx a nivel HT a nivel nacional imputable a cada uno de los servicios *)
Print[TableForm[{invhosttandem, invhosttandemLoc, invhosttandemLDN, invhosttandemLDI, invhosttandemalqA1,
  invhosttandemalqB1, invhosttandemalqC1, invhosttandemalqLL, invhosttandemalqLP, invhosttandemalqAS,
  invhosttandemalqBS, invhosttandemalqCS, invhosttandemalqABC2, invhosttandemalqABCnop, invhosttandemadsl,
  invhosttandemitx, invhosttandemTOTAL}, TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"invhosttandem", "invhosttandemLoc", "invhosttandemLDN", "invhosttandemLDI",
    "invhosttandemalqA1", "invhosttandemalqB1", "invhosttandemalqC1", "invhosttandemalqLL", "invhosttandemalqLP",
    "invhosttandemalqAS", "invhosttandemalqBS", "invhosttandemalqCS", "invhosttandemalqABC2",
    "invhosttandemalqABCnop", "invhosttandemadsl", "invhosttandemitx", "invhosttandemTOTAL"}}]]];
```

```
Print[Style["INVERSIÓN POR SERVICIO – HOST TANDEM",
  16, Bold, White, Underlined, Background → Blue]]];
```

```
Resinvhosttandemx = Table[0, {jx, nx}, {ix, 13}];
Do[Resinvhosttandemx[[ix, 1]] = invhosttandemx[[ix]];
  Resinvhosttandemx[[ix, 2]] = invhosttandemalqxAl[[ix]];
  Resinvhosttandemx[[ix, 3]] = invhosttandemalqxB1[[ix]];
  Resinvhosttandemx[[ix, 4]] = invhosttandemalqxC1[[ix]];
  Resinvhosttandemx[[ix, 5]] = invhosttandemalqxLL[[ix]];
  Resinvhosttandemx[[ix, 6]] = invhosttandemalqxLP[[ix]];
  Resinvhosttandemx[[ix, 7]] = invhosttandemalqxAS[[ix]];
  Resinvhosttandemx[[ix, 8]] = invhosttandemalqxBs[[ix]];
  Resinvhosttandemx[[ix, 9]] = invhosttandemalqxCs[[ix]];
  Resinvhosttandemx[[ix, 10]] = invhosttandemalqxABC2[[ix]];
  Resinvhosttandemx[[ix, 11]] = invhosttandemalqxABCnop[[ix]];
  Resinvhosttandemx[[ix, 12]] = invhosttandemitxx[[ix]];
  Resinvhosttandemx[[ix, 13]] = invhosttandemadslx[[ix]]; , {ix, nx}];
Print[TableForm[Resinvhosttandemx]];
```

PRINT["INVERSIÓN COMPILADA A NIVEL REMOTA-CABECERA"]

```
Print["CosteAñadidoTrxPorDistanciaRHx"]
```

```
CosteAñadidoTrxPorDistanciaRHx=zeror;
```

```
Do[If[deptx[[ix]]≠ LIMA ∧ host1x[[ix]]==0 ∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaRHx[[ix]]=AñadidoTrxPorKm*distance[ix, idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaRHx[[ix]]=CosteAñadidoTrxPorDistanciaRHx[[ix]]*
  If[e1compRHTOTALx[[ix]]≠0 ∧ host1x[[ix]]≠1, (e1compRHx[[ix]]/e1compRHTOTALx[[ix]]), 0];];
, {ix, nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaRHx]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Voz. Nótese que si el número de EIs es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxA1"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxA1=zeror;
```

```
Do[If[deptx[[ix]]≠ LIMA && host1x[[ix]]==0 ∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxA1[[ix]]=AñadidoTrxPorKm*distance[ix, idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxA1[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxA1[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1, (e1compalqRHxA1[[ix]]/e1compRHTOTALx[[ix]]), 0];];
, {ix, nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxA1]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango A. Nótese que si el número de EIs es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxB1"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxB1=zeror;
```

```
Do[If[deptx[[ix]]≠ LIMA && host1x[[ix]]==0 ∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxB1[[ix]]=AñadidoTrxPorKm*distance[ix, idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxB1[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxB1[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1, (e1compalqRHxB1[[ix]]/e1compRHTOTALx[[ix]]), 0];];
, {ix, nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxB1]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango B. Nótese que si el número de EIs es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxC1"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxC1=zeror;
```

```
Do[If[deptx[[ix]]≠ LIMA && host1x[[ix]]==0 ∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxC1[[ix]]=AñadidoTrxPorKm*distance[ix, idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxC1[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxC1[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1, (e1compalqRHxC1[[ix]]/e1compRHTOTALx[[ix]]), 0];];
, {ix, nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxC1]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a

la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango C. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxLL"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxLL=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxLL[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxLL[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxLL[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxLL[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxLL]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Local Lima. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxLP"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxLP=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxLP[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxLP[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxLP[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxLP[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxLP]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Local Provincias. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxAS"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxAS=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxAS[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxAS[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxAS[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxAS[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxAS]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango A Satelital. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxBS"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxBS=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxBS[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxBS[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxBS[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxBS[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxBS]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango B Satelital. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxCS"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxCS=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxCS[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxCS[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxCS[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxCS[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxCS]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos Rango C Satelital. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxABC2"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxABC2=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0^ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxABC2[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxABC2[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxABC2[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxABC2[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxABC2]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo

por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Otros Circuitos Rango A, B y C. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaalqRHxABCnop"]
```

```
CosteAñadidoTrxPorDistanciaalqRHxABCnop=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaalqRHxABCnop[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaalqRHxABCnop[[ix]]=CosteAñadidoTrxPorDistanciaalqRHxABCnop[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compalqRHxABCnop[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaalqRHxABCnop]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos de No Operadores. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaaitxRHx"]
```

```
CosteAñadidoTrxPorDistanciaaitxRHx=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==0∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaaitxRHx[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaaitxRHx[[ix]]=CosteAñadidoTrxPorDistanciaaitxRHx[[ix]]*
  If[e1compRHTOTALx[[ix]]!=0&&host1x[[ix]]!=1,(e1compitxRHx[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaaitxRHx]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de Circuitos de Interconexión. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaaadsLRHx"]
```

```
CosteAñadidoTrxPorDistanciaaadsLRHx=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA ∧ host1x[[ix]]==0∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaaadsLRHx[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaaadsLRHx[[ix]]=CosteAñadidoTrxPorDistanciaaadsLRHx[[ix]]*
  If[e1compRHTOTALx[[ix]]≠0 ∧ host1x[[ix]]≠1,(e1compadsLRHx[[ix]]/e1compRHTOTALx[[ix]]),0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaaadsLRHx]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo del servicio de ADSL. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["CosteAñadidoTrxPorDistanciaRHTOTALx"]
```

```
CosteAñadidoTrxPorDistanciaRHTOTALx=zeror;
```

```
Do[If[depts[[ix]]≠ LIMA ∧ host1x[[ix]]==0∧ transtechx[[ix]] ≠ Radio,
  CosteAñadidoTrxPorDistanciaRHTOTALx[[ix]]=AñadidoTrxPorKm*distance[ix,idinvertx[[CentralRef]]];
  CosteAñadidoTrxPorDistanciaRHTOTALx[[ix]]=CosteAñadidoTrxPorDistanciaRHTOTALx[[ix]]*If[e1compRHTOTALx[[ix]]≠0 ∧
host1x[[ix]]≠1,1,0];];
,{ix,nx}];
```

```
Print[CosteAñadidoTrxPorDistanciaRHTOTALx]
```

(* El costo de traslado de los equipos de Tx a provincias: Si la central no esta en Lima y es cabecera equivale a la distancia de cada central hasta la central de referencia en Lima (Tandem de Lima) multiplicado por el costo por Km. En todas las Remotas entonces el valor es cero. El resultado se multiplica por el respectivo factor de imputación para estimar el costo de todos los servicios. Nótese que si el número de E1s es cero, entonces se multiplica por cero por lo que costo añadido debe ser cero. Ojo, en todas las cabeceras e1compRHTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel RH todo es cero *)

```
Print["invremotehostx1"]
```

```
invremotehostx1=radioinvremotehostx;
```

```
Do[If[fiberx[[ix]],invremotehostx1[[ix]]=fiberinvremotehostx1[[ix]]],{ix,nx}]
```

```
Do[invremotehostx1[[ix]]=invremotehostx1[[ix]]+CosteAñadidoTrxPorDistanciaRHx[[ix]];, {ix,nx}];
```

```
Print[invremotehostx1]
```

(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Voz. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostx. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostx1. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

```
Print["invremotehostLocx1"]
```

```
invremotehostLocx1=zeror;
```

```
Do[If[loadcompRHx[[ix]]>0,invremotehostLocx1[[ix]]=invremotehostx1[[ix]]*loadcompRHLocx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}];
```

```
Print[invremotehostLocx1]
```

(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio Local de Voz. *)

```
Print["invremotehostLDNx1"]
```

```
invremotehostLDNx1=zeror;
```

```
Do[If[loadcompRHx[[ix]]>0,invremotehostLDNx1[[ix]]=invremotehostx1[[ix]]*loadcompRHLDNx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}];
```

```
Print[invremotehostLDNx1]
```

(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio LDN de Voz. *)

```
Print["invremotehostLDIx1"]
```



```

invremotehostLDIx1=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLDIx1[[ix]]=invremotehostx1[[ix]]*loadcompRHLDIx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}];
Print[invremotehostLDIx1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio LDI de Voz. *)

Print["invremotehostalqx1A1"]
invremotehostalqx1A1=radioinvremotehostalqx1A1;
Do[If[fiberx[[ix]],invremotehostalqx1A1[[ix]]=fiberinvremotehostalqx1A1[[ix]]],{ix,nx}]
Do[invremotehostalqx1A1[[ix]]=invremotehostalqx1A1[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxA1[[ix]]],{ix,nx}];
Print[invremotehostalqx1A1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango A. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1A1. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1A1. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1B1"]
invremotehostalqx1B1=radioinvremotehostalqx1B1;
Do[If[fiberx[[ix]],invremotehostalqx1B1[[ix]]=fiberinvremotehostalqx1B1[[ix]]],{ix,nx}]
Do[invremotehostalqx1B1[[ix]]=invremotehostalqx1B1[[ix]]+CosteAñadidoTrxPorDistanciaalqRHB1[[ix]]],{ix,nx}];
Print[invremotehostalqx1B1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango B. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1B1. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1B1. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1C1"]
invremotehostalqx1C1=radioinvremotehostalqx1C1;
Do[If[fiberx[[ix]],invremotehostalqx1C1[[ix]]=fiberinvremotehostalqx1C1[[ix]]],{ix,nx}]
Do[invremotehostalqx1C1[[ix]]=invremotehostalqx1C1[[ix]]+CosteAñadidoTrxPorDistanciaalqRHC1[[ix]]],{ix,nx}];
Print[invremotehostalqx1C1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango C. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1C1. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1C1. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1LL"]
invremotehostalqx1LL=radioinvremotehostalqx1LL;
Do[If[fiberx[[ix]],invremotehostalqx1LL[[ix]]=fiberinvremotehostalqx1LL[[ix]]],{ix,nx}]
Do[invremotehostalqx1LL[[ix]]=invremotehostalqx1LL[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxLL[[ix]]],{ix,nx}];
Print[invremotehostalqx1LL]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Local Lima. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1LL. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1LL. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1LP"]
invremotehostalqx1LP=radioinvremotehostalqx1LP;
Do[If[fiberx[[ix]],invremotehostalqx1LP[[ix]]=fiberinvremotehostalqx1LP[[ix]]],{ix,nx}]
Do[invremotehostalqx1LP[[ix]]=invremotehostalqx1LP[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxLP[[ix]]],{ix,nx}];
Print[invremotehostalqx1LP]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Local Provincias. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1LP. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1LP. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1AS"]
invremotehostalqx1AS=radioinvremotehostalqx1AS;
Do[If[fiberx[[ix]],invremotehostalqx1AS[[ix]]=fiberinvremotehostalqx1AS[[ix]]],{ix,nx}]
Do[invremotehostalqx1AS[[ix]]=invremotehostalqx1AS[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxAS[[ix]]],{ix,nx}];
Print[invremotehostalqx1AS]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango A Satelital. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1AS. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1AS. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1BS"]
invremotehostalqx1BS=radioinvremotehostalqx1BS;
Do[If[fiberx[[ix]],invremotehostalqx1BS[[ix]]=fiberinvremotehostalqx1BS[[ix]]],{ix,nx}]
Do[invremotehostalqx1BS[[ix]]=invremotehostalqx1BS[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxBS[[ix]]],{ix,nx}];
Print[invremotehostalqx1BS]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango B Satelital. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1BS. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada
una de las centrales del departamento será fiberinvremotehostalqx1BS. Luego, identificado el nivel de inversión a
tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

Print["invremotehostalqx1CS"]
invremotehostalqx1CS=radioinvremotehostalqx1CS;
Do[If[fiberx[[ix]],invremotehostalqx1CS[[ix]]=fiberinvremotehostalqx1CS[[ix]]],{ix,nx}]
Do[invremotehostalqx1CS[[ix]]=invremotehostalqx1CS[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxCS[[ix]]],{ix,nx}];
Print[invremotehostalqx1CS]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos
Rango C Satelital. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del
departamento será radioinvremotehostalqx1CS. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada

```

una de las centrales del departamento será fiberinvremotehostalqx1CS. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)

```
Print["invremotehostalqx1ABC2"]
invremotehostalqx1ABC2=radioinvremotehostalqxABC2;
Do[If[fiberx[[ix]],invremotehostalqx1ABC2[[ix]]=fiberinvremotehostalqx1ABC2[[ix]]],{ix,nx}]
Do[invremotehostalqx1ABC2[[ix]]=invremotehostalqx1ABC2[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxABC2[[ix]];, {ix,nx}]
;
Print[invremotehostalqx1ABC2]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Otros Circuitos Rango A, B y C. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostalqxABC2. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostalqx1ABC2. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)
```

```
Print["invremotehostalqx1ABCnop"]
invremotehostalqx1ABCnop=radioinvremotehostalqxABCnop;
Do[If[fiberx[[ix]],invremotehostalqx1ABCnop[[ix]]=fiberinvremotehostalqx1ABCnop[[ix]]],{ix,nx}]
Do[invremotehostalqx1ABCnop[[ix]]=invremotehostalqx1ABCnop[[ix]]+CosteAñadidoTrxPorDistanciaalqRHxABCnop[[ix]];, {ix,nx}];
Print[invremotehostalqx1ABCnop]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos de No Operadores. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostalqxB1nop. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostalqx1B1nop. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)
```

```
Print["invremotehostitxx1"]
invremotehostitxx1=radioinvremotehostitxx;
Do[If[fiberx[[ix]],invremotehostitxx1[[ix]]=fiberinvremotehostitxx1[[ix]]],{ix,nx}]
Do[invremotehostitxx1[[ix]]=invremotehostitxx1[[ix]]+CosteAñadidoTrxPorDistanciaaitxRHx[[ix]];, {ix,nx}];
Print[invremotehostitxx1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de Circuitos de Interconexión. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostitxx. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostitxx1. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)
```

```
Print["invremotehostadslx1"]
invremotehostadslx1=radioinvremotehostadslx;
Do[If[fiberx[[ix]],invremotehostadslx1[[ix]]=fiberinvremotehostadslx1[[ix]]],{ix,nx}]
Do[invremotehostadslx1[[ix]]=invremotehostadslx1[[ix]]+CosteAñadidoTrxPorDistanciaadslRHx[[ix]];, {ix,nx}];
Print[invremotehostadslx1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable al servicio de ADSL. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostadslx. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostadslx1. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)
```

```
Print["invremotehostTOTALx1"]
invremotehostTOTALx1=radioinvremotehostTOTALx;
Do[If[fiberx[[ix]],invremotehostTOTALx1[[ix]]=fiberinvremotehostTOTALx1[[ix]]],{ix,nx}]
Do[invremotehostTOTALx1[[ix]]=invremotehostTOTALx1[[ix]]+CosteAñadidoTrxPorDistanciaRHTOTALx[[ix]];, {ix,nx}];
Print[invremotehostTOTALx1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel RH imputable a todos los servicios. Utiliza la inversión en radio, el nivel de inversión válido para cada una de las centrales del departamento será radioinvremotehostTOTALx. Luego, si la Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será fiberinvremotehostTOTALx1. Luego, identificado el nivel de inversión a tomar en cuenta en cada central, se añade el costo añadido el traslado de los equipos desde Lima a provincias *)
```

```
Print["invremotehostd1"]
invremotehostd1=zerod;
Do[invremotehostd1[[deptx[[ix]]]]=invremotehostd1[[deptx[[ix]]]]+invremotehostx1[[ix]],{ix,nx}]
Print[invremotehostd1]
(* Acumula a nivel departamental la variable invremotehostx1 imputable al servicio de Voz *)
```

```
Print["invremotehostLocd1"]
invremotehostLocd1=zerod;
Do[invremotehostLocd1[[deptx[[ix]]]]=invremotehostLocd1[[deptx[[ix]]]]+invremotehostLocx1[[ix]],{ix,nx}]
Print[invremotehostLocd1]
(* Acumula a nivel departamental la variable invremotehostLocx1 imputable al servicio Local de Voz *)
```

```
Print["invremotehostLDNd1"]
invremotehostLDNd1=zerod;
Do[invremotehostLDNd1[[deptx[[ix]]]]=invremotehostLDNd1[[deptx[[ix]]]]+invremotehostLDNx1[[ix]],{ix,nx}]
Print[invremotehostLDNd1]
(* Acumula a nivel departamental la variable invremotehostLDNx1 imputable al servicio LDN de Voz *)
```

```
Print["invremotehostLDId1"]
invremotehostLDId1=zerod;
Do[invremotehostLDId1[[deptx[[ix]]]]=invremotehostLDId1[[deptx[[ix]]]]+invremotehostLDIx1[[ix]],{ix,nx}]
Print[invremotehostLDId1]
(* Acumula a nivel departamental la variable invremotehostLDIx1 imputable al servicio LDI de Voz *)
```

```
Print["invremotehostalqd1A1"]
invremotehostalqd1A1=zerod;
Do[invremotehostalqd1A1[[deptx[[ix]]]]=invremotehostalqd1A1[[deptx[[ix]]]]+invremotehostalqx1A1[[ix]],{ix,nx}]
Print[invremotehostalqd1A1]
(* Acumula a nivel departamental la variable invremotehostalqx1A1 imputable al servicio de Circuitos Rango A *)
```

```

Print["invremotehostald1B1"]
invremotehostald1B1=zerod;
Do[invremotehostald1B1[[deptx[[ix]]]]=invremotehostald1B1[[deptx[[ix]]]]+invremotehostald1B1[[ix]],{ix,nx}]
Print[invremotehostald1B1]
(* Acumula a nivel departamental la variable invremotehostald1B1 imputable al servicio de Circuitos Rango B *)

Print["invremotehostald1C1"]
invremotehostald1C1=zerod;
Do[invremotehostald1C1[[deptx[[ix]]]]=invremotehostald1C1[[deptx[[ix]]]]+invremotehostald1C1[[ix]],{ix,nx}]
Print[invremotehostald1C1]
(* Acumula a nivel departamental la variable invremotehostald1C1 imputable al servicio de Circuitos Rango C *)

Print["invremotehostald1LL"]
invremotehostald1LL=zerod;
Do[invremotehostald1LL[[deptx[[ix]]]]=invremotehostald1LL[[deptx[[ix]]]]+invremotehostald1LL[[ix]],{ix,nx}]
Print[invremotehostald1LL]
(* Acumula a nivel departamental la variable invremotehostald1LL imputable al servicio de Circuitos Local Lima *)

Print["invremotehostald1LP"]
invremotehostald1LP=zerod;
Do[invremotehostald1LP[[deptx[[ix]]]]=invremotehostald1LP[[deptx[[ix]]]]+invremotehostald1LP[[ix]],{ix,nx}]
Print[invremotehostald1LP]
(* Acumula a nivel departamental la variable invremotehostald1LP imputable al servicio de Circuitos Local Provincias *)

Print["invremotehostald1AS"]
invremotehostald1AS=zerod;
Do[invremotehostald1AS[[deptx[[ix]]]]=invremotehostald1AS[[deptx[[ix]]]]+invremotehostald1AS[[ix]],{ix,nx}]
Print[invremotehostald1AS]
(* Acumula a nivel departamental la variable invremotehostald1AS imputable al servicio de Circuitos Rango A Satelital*)

Print["invremotehostald1BS"]
invremotehostald1BS=zerod;
Do[invremotehostald1BS[[deptx[[ix]]]]=invremotehostald1BS[[deptx[[ix]]]]+invremotehostald1BS[[ix]],{ix,nx}]
Print[invremotehostald1BS]
(* Acumula a nivel departamental la variable invremotehostald1BS imputable al servicio de Circuitos Rango B Satelital*)

Print["invremotehostald1CS"]
invremotehostald1CS=zerod;
Do[invremotehostald1CS[[deptx[[ix]]]]=invremotehostald1CS[[deptx[[ix]]]]+invremotehostald1CS[[ix]],{ix,nx}]
Print[invremotehostald1CS]
(* Acumula a nivel departamental la variable invremotehostald1CS imputable al servicio de Circuitos Rango C Satelital*)

Print["invremotehostald1ABC2"]
invremotehostald1ABC2=zerod;
Do[invremotehostald1ABC2[[deptx[[ix]]]]=invremotehostald1ABC2[[deptx[[ix]]]]+invremotehostald1ABC2[[ix]],{ix,nx}]
Print[invremotehostald1ABC2]
(* Acumula a nivel departamental la variable invremotehostald1ABC2 imputable al servicio de Otros Circuitos Rango A, B y C *)

Print["invremotehostald1ABCnop"]
invremotehostald1ABCnop=zerod;
Do[invremotehostald1ABCnop[[deptx[[ix]]]]=invremotehostald1ABCnop[[deptx[[ix]]]]+invremotehostald1ABCnop[[ix]],{ix,nx}]
Print[invremotehostald1ABCnop]
(* Acumula a nivel departamental la variable invremotehostald1ABCnop imputable al servicio de Circuitos de No Operadores*)

Print["invremotehostitxd1"]
invremotehostitxd1=zerod;
Do[invremotehostitxd1[[deptx[[ix]]]]=invremotehostitxd1[[deptx[[ix]]]]+invremotehostitxd1[[ix]],{ix,nx}]
Print[invremotehostitxd1]
(* Acumula a nivel departamental la variable invremotehostitxd1 imputable al servicio de Circuitos de Interconexión *)

Print["invremotehostadsl1"]
invremotehostadsl1=zerod;
Do[invremotehostadsl1[[deptx[[ix]]]]=invremotehostadsl1[[deptx[[ix]]]]+invremotehostadsl1[[ix]],{ix,nx}]
Print[invremotehostadsl1]
(* Acumula a nivel departamental la variable invremotehostadsl1 imputable al servicio de ADSL *)

Print["invremotehostTOTALd1"]
invremotehostTOTALd1=zerod;
Do[invremotehostTOTALd1[[deptx[[ix]]]]=invremotehostTOTALd1[[deptx[[ix]]]]+invremotehostTOTALd1[[ix]],{ix,nx}]
Print[invremotehostTOTALd1]
(* Acumula a nivel departamental la variable invremotehostTOTALd1 imputable a todos los servicios *)

Print["invremotehostx2"]
invremotehostx2=zerod;
Do[If[fiberx[[ix]],invremotehostx2[[ix]]=fiberinvremotehostx2[[ix]],{ix,nx}]
Do[If[deptx[[ix]]!= LIMA && transtechx[[ix]]==Enterrado,
  invremotehostx2[[ix]]=0.55*invremotehostx2[[ix]]];,{ix,nx}];
Print[invremotehostx2]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Voz. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostx2, en todos los demás casos es cero. El resultado se

```

```

multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostLocx2"]
invremotehostLocx2=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLocx2[[ix]]=invremotehostx2[[ix]]*loadcompRHLocx[[ix]]/loadcompRHx[[ix]]],{ix,nx}];
Print[invremotehostLocx2]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio Local de Voz. *)

Print["invremotehostLDNx2"]
invremotehostLDNx2=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLDNx2[[ix]]=invremotehostx2[[ix]]*loadcompRHLDNx[[ix]]/loadcompRHx[[ix]]],{ix,nx}];
Print[invremotehostLDNx2]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio LDN de Voz. *)

Print["invremotehostLDIx2"]
invremotehostLDIx2=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLDIx2[[ix]]=invremotehostx2[[ix]]*loadcompRHLDIx[[ix]]/loadcompRHx[[ix]]],{ix,nx}];
Print[invremotehostLDIx2]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio LDI de Voz. *)

Print["invremotehostalqx2A1"]
invremotehostalqx2A1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2A1[[ix]]=fiberinvremotehostalqx2A1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2A1[[ix]]=0.55*invremotehostalqx2A1[[ix]]];,{ix,nx}];
Print[invremotehostalqx2A1]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango A. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2A1, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx2B1"]
invremotehostalqx2B1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2B1[[ix]]=fiberinvremotehostalqx2B1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA ∧ transtechx[[ix]]==Enterrado,
  invremotehostalqx2B1[[ix]]=0.55*invremotehostalqx2B1[[ix]]];,{ix,nx}];
Print[invremotehostalqx2B1]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango B. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2B1, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx2C1"]
invremotehostalqx2C1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2C1[[ix]]=fiberinvremotehostalqx2C1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA ∧ transtechx[[ix]]==Enterrado,
  invremotehostalqx2C1[[ix]]=0.55*invremotehostalqx2C1[[ix]]];,{ix,nx}];
Print[invremotehostalqx2C1]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango C. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2C1, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx2LL"]
invremotehostalqx2LL=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2LL[[ix]]=fiberinvremotehostalqx2LL[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2LL[[ix]]=0.55*invremotehostalqx2LL[[ix]]];,{ix,nx}];
Print[invremotehostalqx2LL]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Local Lima. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2LL, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx2LP"]
invremotehostalqx2LP=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2LP[[ix]]=fiberinvremotehostalqx2LP[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2LP[[ix]]=0.55*invremotehostalqx2LP[[ix]]];,{ix,nx}];
Print[invremotehostalqx2LP]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Local Lima. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2LL, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx2AS"]
invremotehostalqx2AS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2AS[[ix]]=fiberinvremotehostalqx2AS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2AS[[ix]]=0.55*invremotehostalqx2AS[[ix]]];,{ix,nx}];
Print[invremotehostalqx2AS]
(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango A

```


Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2AS, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostalqx2BS"]
invremotehostalqx2BS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2BS[[ix]]=fiberinvremotehostalqx2BS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2BS[[ix]]=0.55*invremotehostalqx2BS[[ix]]];,{ix,nx}];
Print[invremotehostalqx2BS]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango B Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2BS, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostalqx2CS"]
invremotehostalqx2CS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2CS[[ix]]=fiberinvremotehostalqx2CS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2CS[[ix]]=0.55*invremotehostalqx2CS[[ix]]];,{ix,nx}];
Print[invremotehostalqx2CS]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos Rango C Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2CS, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostalqx2ABC2"]
invremotehostalqx2ABC2=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2ABC2[[ix]]=fiberinvremotehostalqx2ABC2[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2ABC2[[ix]]=0.55*invremotehostalqx2ABC2[[ix]]];,{ix,nx}];
Print[invremotehostalqx2ABC2]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Otros Circuitos Rango A, B y C. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2ABC2, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostalqx2ABCnop"]
invremotehostalqx2ABCnop=zerox;
Do[If[fiberx[[ix]],invremotehostalqx2ABCnop[[ix]]=fiberinvremotehostalqx2ABCnop[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx2ABCnop[[ix]]=0.55*invremotehostalqx2ABCnop[[ix]]];,{ix,nx}];
Print[invremotehostalqx2ABCnop]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos de No Operadores. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx2B1nop, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostitxx2"]
invremotehostitxx2=zerox;
Do[If[fiberx[[ix]],invremotehostitxx2[[ix]]=fiberinvremotehostitxx2[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostitxx2[[ix]]=0.55*invremotehostitxx2[[ix]]];,{ix,nx}];
Print[invremotehostitxx2]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de Circuitos de Interconexión. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostitxx2, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostadslx2"]
invremotehostadslx2=zerox;
Do[If[fiberx[[ix]],invremotehostadslx2[[ix]]=fiberinvremotehostadslx2[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA ^ transtechx[[ix]]==Enterrado,
  invremotehostadslx2[[ix]]=0.55*invremotehostadslx2[[ix]]];,{ix,nx}];
Print[invremotehostadslx2]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable al servicio de ADSL. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostadslx2, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostTOTALx2"]
invremotehostTOTALx2=zerox;
Do[If[fiberx[[ix]],invremotehostTOTALx2[[ix]]=fiberinvremotehostTOTALx2[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA ^ transtechx[[ix]]==Enterrado,
  invremotehostTOTALx2[[ix]]=0.55*invremotehostTOTALx2[[ix]]];,{ix,nx}];
Print[invremotehostTOTALx2]
```

(* Compila o resume el total de inversiones aplicable a los cables imputable a todos los servicios. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se considera el nivel de inversión fiberinvremotehostTOTALx2, en todos los demás casos es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red fija se definió que el 45% se comparte) *)

```
Print["invremotehostd2"]
invremotehostd2=zerod;
```

```

Do[invremotehostd2[[deptx[[ix]]]]=invremotehostd2[[deptx[[ix]]]+invremotehostx2[[ix]],{ix,nx}]
Print[invremotehostd2]
(* Acumula a nivel departamental la variable invremotehostx2 imputable al servicio de Voz *)

Print["invremotehostLocd2"]
invremotehostLocd2=zerod;
Do[invremotehostLocd2[[deptx[[ix]]]]=invremotehostLocd2[[deptx[[ix]]]+invremotehostLocx2[[ix]],{ix,nx}]
Print[invremotehostLocd2]
(* Acumula a nivel departamental la variable invremotehostLocx2 imputable al servicio Local de Voz *)

Print["invremotehostLDNd2"]
invremotehostLDNd2=zerod;
Do[invremotehostLDNd2[[deptx[[ix]]]]=invremotehostLDNd2[[deptx[[ix]]]+invremotehostLDNx2[[ix]],{ix,nx}]
Print[invremotehostLDNd2]
(* Acumula a nivel departamental la variable invremotehostLDNx2 imputable al servicio LDN de Voz *)

Print["invremotehostLDId2"]
invremotehostLDId2=zerod;
Do[invremotehostLDId2[[deptx[[ix]]]]=invremotehostLDId2[[deptx[[ix]]]+invremotehostLDIx2[[ix]],{ix,nx}]
Print[invremotehostLDId2]
(* Acumula a nivel departamental la variable invremotehostLDIx2 imputable al servicio LDI de Voz *)

Print["invremotehostalqd2A1"]
invremotehostalqd2A1=zerod;
Do[invremotehostalqd2A1[[deptx[[ix]]]]=invremotehostalqd2A1[[deptx[[ix]]]+invremotehostalqx2A1[[ix]],{ix,nx}]
Print[invremotehostalqd2A1]
(* Acumula a nivel departamental la variable invremotehostalqx2A1 imputable al servicio de Circuitos Rango A *)

Print["invremotehostalqd2B1"]
invremotehostalqd2B1=zerod;
Do[invremotehostalqd2B1[[deptx[[ix]]]]=invremotehostalqd2B1[[deptx[[ix]]]+invremotehostalqx2B1[[ix]],{ix,nx}]
Print[invremotehostalqd2B1]
(* Acumula a nivel departamental la variable invremotehostalqx2B1 imputable al servicio de Circuitos Rango B *)

Print["invremotehostalqd2C1"]
invremotehostalqd2C1=zerod;
Do[invremotehostalqd2C1[[deptx[[ix]]]]=invremotehostalqd2C1[[deptx[[ix]]]+invremotehostalqx2C1[[ix]],{ix,nx}]
Print[invremotehostalqd2C1]
(* Acumula a nivel departamental la variable invremotehostalqx2C1 imputable al servicio de Circuitos Rango C *)

Print["invremotehostalqd2LL"]
invremotehostalqd2LL=zerod;
Do[invremotehostalqd2LL[[deptx[[ix]]]]=invremotehostalqd2LL[[deptx[[ix]]]+invremotehostalqx2LL[[ix]],{ix,nx}]
Print[invremotehostalqd2LL]
(* Acumula a nivel departamental la variable invremotehostalqx2LL imputable al servicio de Circuitos Local Lima *)

Print["invremotehostalqd2LP"]
invremotehostalqd2LP=zerod;
Do[invremotehostalqd2LP[[deptx[[ix]]]]=invremotehostalqd2LP[[deptx[[ix]]]+invremotehostalqx2LP[[ix]],{ix,nx}]
Print[invremotehostalqd2LP]
(* Acumula a nivel departamental la variable invremotehostalqx2LP imputable al servicio de Circuitos Local Provincias *)

Print["invremotehostalqd2AS"]
invremotehostalqd2AS=zerod;
Do[invremotehostalqd2AS[[deptx[[ix]]]]=invremotehostalqd2AS[[deptx[[ix]]]+invremotehostalqx2AS[[ix]],{ix,nx}]
Print[invremotehostalqd2AS]
(* Acumula a nivel departamental la variable invremotehostalqx2AS imputable al servicio de Circuitos Rango A Satelital*)

Print["invremotehostalqd2BS"]
invremotehostalqd2BS=zerod;
Do[invremotehostalqd2BS[[deptx[[ix]]]]=invremotehostalqd2BS[[deptx[[ix]]]+invremotehostalqx2BS[[ix]],{ix,nx}]
Print[invremotehostalqd2BS]
(* Acumula a nivel departamental la variable invremotehostalqx2BS imputable al servicio de Circuitos Rango B Satelital*)

Print["invremotehostalqd2CS"]
invremotehostalqd2CS=zerod;
Do[invremotehostalqd2CS[[deptx[[ix]]]]=invremotehostalqd2CS[[deptx[[ix]]]+invremotehostalqx2CS[[ix]],{ix,nx}]
Print[invremotehostalqd2CS]
(* Acumula a nivel departamental la variable invremotehostalqx2CS imputable al servicio de Circuitos Rango C Satelital*)

Print["invremotehostalqd2ABC2"]
invremotehostalqd2ABC2=zerod;
Do[invremotehostalqd2ABC2[[deptx[[ix]]]]=invremotehostalqd2ABC2[[deptx[[ix]]]+invremotehostalqx2ABC2[[ix]],{ix,nx}]
Print[invremotehostalqd2ABC2]
(* Acumula a nivel departamental la variable invremotehostalqx2ABC2 imputable al servicio de Otros Circuitos Rango A, B y C*)

Print["invremotehostalqd2ABCnop"]
invremotehostalqd2ABCnop=zerod;
Do[invremotehostalqd2ABCnop[[deptx[[ix]]]]=invremotehostalqd2ABCnop[[deptx[[ix]]]+invremotehostalqx2ABCnop[[ix]],{ix,nx}]
Print[invremotehostalqd2ABCnop]
(* Acumula a nivel departamental la variable invremotehostalqx2ABCnop imputable al servicio de Circuitos de No Operadores*)

Print["invremotehostitxd2"]

```



```

invremotehostitxd2=zerod;
Do[invremotehostitxd2[[deptx[[ix]]]]=invremotehostitxd2[[deptx[[ix]]]]+invremotehostitxx2[[ix]],{ix,nx}]
Print[invremotehostitxd2]
(* Acumula a nivel departamental la variable invremotehostitxx2 imputable al servicio de Circuitos de
Interconexión *)

Print["invremotehostadsld2"]
invremotehostadsld2=zerod;
Do[invremotehostadsld2[[deptx[[ix]]]]=invremotehostadsld2[[deptx[[ix]]]]+invremotehostadslx2[[ix]],{ix,nx}]
Print[invremotehostadsld2]
(* Acumula a nivel departamental la variable invremotehostadslx2 imputable al servicio de ADSL *)

Print["invremotehostTOTALd2"]
invremotehostTOTALd2=zerod;
Do[invremotehostTOTALd2[[deptx[[ix]]]]=invremotehostTOTALd2[[deptx[[ix]]]]+invremotehostTOTALx2[[ix]],{ix,nx}]
Print[invremotehostTOTALd2]
(* Acumula a nivel departamental la variable invremotehostTOTALx2 imputable a todos los servicios*)

Print["invremotehostx3"]
invremotehostx3=zerox;
Do[If[fiberx[[ix]],invremotehostx3[[ix]]=fiberinvremotehostx3[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostx3[[ix]]=0.55*invremotehostx3[[ix]]];,{ix,nx}];
Print[invremotehostx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Voz. Como dicha
información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se
considera el nivel de inversión fiberinvremotehostx3, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostLocx3"]
invremotehostLocx3=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLocx3[[ix]]=invremotehostx3[[ix]]*loadcompRHLocx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}]
Print[invremotehostLocx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio Local de Voz.*)

Print["invremotehostLDNx3"]
invremotehostLDNx3=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLDNx3[[ix]]=invremotehostx3[[ix]]*loadcompRHLDNx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}]
Print[invremotehostLDNx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio LDN de Voz.*)

Print["invremotehostLDIx3"]
invremotehostLDIx3=zerox;
Do[If[loadcompRHx[[ix]]>0,invremotehostLDIx3[[ix]]=invremotehostx3[[ix]]*loadcompRHLDIx[[ix]]/loadcompRHx[[ix]]],{
ix,nx}]
Print[invremotehostLDIx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio LDI de Voz.*)

Print["invremotehostalqx3A1"]
invremotehostalqx3A1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3A1[[ix]]=fiberinvremotehostalqx3A1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3A1[[ix]]=0.55*invremotehostalqx3A1[[ix]]];,{ix,nx}];
Print[invremotehostalqx3A1]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango A.
Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra
se considera el nivel de inversión fiberinvremotehostalqx3A1, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3B1"]
invremotehostalqx3B1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3B1[[ix]]=fiberinvremotehostalqx3B1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3B1[[ix]]=0.55*invremotehostalqx3B1[[ix]]];,{ix,nx}];
Print[invremotehostalqx3B1]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango B.
Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra
se considera el nivel de inversión fiberinvremotehostalqx3B1, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3C1"]
invremotehostalqx3C1=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3C1[[ix]]=fiberinvremotehostalqx3C1[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3C1[[ix]]=0.55*invremotehostalqx3C1[[ix]]];,{ix,nx}];
Print[invremotehostalqx3C1]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango C.
Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra
se considera el nivel de inversión fiberinvremotehostalqx3C1, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3LL"]
invremotehostalqx3LL=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3LL[[ix]]=fiberinvremotehostalqx3LL[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,

```

```
invremotehostalqx3LL[[ix]]=0.55*invremotehostalqx3LL[[ix]];;{ix,nx}};
Print[invremotehostalqx3LL]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Local
Lima. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es
fibra se considera el nivel de inversión fiberinvremotehostalqx3LL, en todos los demás casos es cero. El
resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el
modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3LP"]
invremotehostalqx3LP=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3LP[[ix]]=fiberinvremotehostalqx3LP[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3LP[[ix]]=0.55*invremotehostalqx3LP[[ix]];;{ix,nx}};
Print[invremotehostalqx3LP]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Local
Provincias. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3LP, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3AS"]
invremotehostalqx3AS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3AS[[ix]]=fiberinvremotehostalqx3AS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3AS[[ix]]=0.55*invremotehostalqx3AS[[ix]];;{ix,nx}};
Print[invremotehostalqx3AS]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango A
Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3AS, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3BS"]
invremotehostalqx3BS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3BS[[ix]]=fiberinvremotehostalqx3BS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3BS[[ix]]=0.55*invremotehostalqx3BS[[ix]];;{ix,nx}};
Print[invremotehostalqx3BS]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango B
Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3BS, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3CS"]
invremotehostalqx3CS=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3CS[[ix]]=fiberinvremotehostalqx3CS[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3CS[[ix]]=0.55*invremotehostalqx3CS[[ix]];;{ix,nx}};
Print[invremotehostalqx3CS]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos Rango C
Satelital. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3CS, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3ABC2"]
invremotehostalqx3ABC2=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3ABC2[[ix]]=fiberinvremotehostalqx3ABC2[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3ABC2[[ix]]=0.55*invremotehostalqx3ABC2[[ix]];;{ix,nx}};
Print[invremotehostalqx3ABC2]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Otros Circuitos
Rango A, B y C. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3ABC2, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostalqx3ABCnop"]
invremotehostalqx3ABCnop=zerox;
Do[If[fiberx[[ix]],invremotehostalqx3ABCnop[[ix]]=fiberinvremotehostalqx3ABCnop[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostalqx3ABCnop[[ix]]=0.55*invremotehostalqx3ABCnop[[ix]];;{ix,nx}};
Print[invremotehostalqx3ABCnop]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos de No
Operadores. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostalqx3ABCnop, en todos los demás casos
es cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)

Print["invremotehostitxx3"]
invremotehostitxx3=zerox;
Do[If[fiberx[[ix]],invremotehostitxx3[[ix]]=fiberinvremotehostitxx3[[ix]]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostitxx3[[ix]]=0.55*invremotehostitxx3[[ix]];;{ix,nx}};
Print[invremotehostitxx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de Circuitos de
Interconexión. Como dicha información sólo es aplicable para el uso de fibra, simplemente si la Tandem del
departamento es fibra se considera el nivel de inversión fiberinvremotehostitxx3, en todos los demás casos es
cero. El resultado se multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima
(En el modelo de la red fija se definió que el 45% se comparte) *)
```

```

Print["invremotehostadslx3"]
invremotehostadslx3=zerod;
Do[If[fiberx[[ix]],invremotehostadslx3[[ix]]=fiberinvremotehostadslx3[[ix]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA && transtechx[[ix]]==Enterrado,
  invremotehostadslx3[[ix]]=0.55*invremotehostadslx3[[ix]]];,{ix,nx}];
Print[invremotehostadslx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable al servicio de ADSL. Como dicha
información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se
considera el nivel de inversión fiberinvremotehostadslx3, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostTOTALx3"]
invremotehostTOTALx3=zerod;
Do[If[fiberx[[ix]],invremotehostTOTALx3[[ix]]=fiberinvremotehostTOTALx3[[ix]],{ix,nx}]
Do[If[deptx[[ix]]≠ LIMA ^ transtechx[[ix]]==Enterrado,
  invremotehostTOTALx3[[ix]]=0.55*invremotehostTOTALx3[[ix]]];,{ix,nx}];
Print[invremotehostTOTALx3]
(* Compila o resume el total de inversiones aplicable a obras civiles imputable a todos los servicios. Como dicha
información sólo es aplicable para el uso de fibra, simplemente si la Tandem del departamento es fibra se
considera el nivel de inversión fiberinvremotehostTOTALx3, en todos los demás casos es cero. El resultado se
multiplica por el factor de compartición de fibra enterrada para los nodos fuera de Lima (En el modelo de la red
fija se definió que el 45% se comparte) *)

Print["invremotehostd3"]
invremotehostd3=zerod;
Do[invremotehostd3[[deptx[[ix]]]]=invremotehostd3[[deptx[[ix]]]]+invremotehostx3[[ix]],{ix,nx}]
Print[invremotehostd3]
(* Acumula a nivel departamental la variable invremotehostx3 imputable al servicio de Voz*)

Print["invremotehostLocd3"]
invremotehostLocd3=zerod;
Do[invremotehostLocd3[[deptx[[ix]]]]=invremotehostLocd3[[deptx[[ix]]]]+invremotehostLocx3[[ix]],{ix,nx}]
Print[invremotehostLocd3]
(* Acumula a nivel departamental la variable invremotehostLocx3 imputable al servicio Local de Voz*)

Print["invremotehostLDNd3"]
invremotehostLDNd3=zerod;
Do[invremotehostLDNd3[[deptx[[ix]]]]=invremotehostLDNd3[[deptx[[ix]]]]+invremotehostLDNx3[[ix]],{ix,nx}]
Print[invremotehostLDNd3]
(* Acumula a nivel departamental la variable invremotehostLDNx3 imputable al servicio LDN de Voz*)

Print["invremotehostLDId3"]
invremotehostLDId3=zerod;
Do[invremotehostLDId3[[deptx[[ix]]]]=invremotehostLDId3[[deptx[[ix]]]]+invremotehostLDIx3[[ix]],{ix,nx}]
Print[invremotehostLDId3]
(* Acumula a nivel departamental la variable invremotehostLDIx3 imputable al servicio LDI de Voz*)

Print["invremotehostalqd3A1"]
invremotehostalqd3A1=zerod;
Do[invremotehostalqd3A1[[deptx[[ix]]]]=invremotehostalqd3A1[[deptx[[ix]]]]+invremotehostalqx3A1[[ix]],{ix,nx}]
Print[invremotehostalqd3A1]
(* Acumula a nivel departamental la variable invremotehostalqx3A1 imputable al servicio de Circuitos Rango A *)

Print["invremotehostalqd3B1"]
invremotehostalqd3B1=zerod;
Do[invremotehostalqd3B1[[deptx[[ix]]]]=invremotehostalqd3B1[[deptx[[ix]]]]+invremotehostalqx3B1[[ix]],{ix,nx}]
Print[invremotehostalqd3B1]
(* Acumula a nivel departamental la variable invremotehostalqx3B1 imputable al servicio de Circuitos Rango B *)

Print["invremotehostalqd3C1"]
invremotehostalqd3C1=zerod;
Do[invremotehostalqd3C1[[deptx[[ix]]]]=invremotehostalqd3C1[[deptx[[ix]]]]+invremotehostalqx3C1[[ix]],{ix,nx}]
Print[invremotehostalqd3C1]
(* Acumula a nivel departamental la variable invremotehostalqx3C1 imputable al servicio de Circuitos Rango C *)

Print["invremotehostalqd3LL"]
invremotehostalqd3LL=zerod;
Do[invremotehostalqd3LL[[deptx[[ix]]]]=invremotehostalqd3LL[[deptx[[ix]]]]+invremotehostalqx3LL[[ix]],{ix,nx}]
Print[invremotehostalqd3LL]
(* Acumula a nivel departamental la variable invremotehostalqx3LL imputable al servicio de Circuitos Local Lima *)

Print["invremotehostalqd3LP"]
invremotehostalqd3LP=zerod;
Do[invremotehostalqd3LP[[deptx[[ix]]]]=invremotehostalqd3LP[[deptx[[ix]]]]+invremotehostalqx3LP[[ix]],{ix,nx}]
Print[invremotehostalqd3LP]
(* Acumula a nivel departamental la variable invremotehostalqx3LP imputable al servicio de Circuitos Local
Provincias *)

Print["invremotehostalqd3AS"]
invremotehostalqd3AS=zerod;
Do[invremotehostalqd3AS[[deptx[[ix]]]]=invremotehostalqd3AS[[deptx[[ix]]]]+invremotehostalqx3AS[[ix]],{ix,nx}]
Print[invremotehostalqd3AS]
(* Acumula a nivel departamental la variable invremotehostalqx3AS imputable al servicio de Circuitos Rango A
Satelital*)

Print["invremotehostalqd3BS"]
invremotehostalqd3BS=zerod;
Do[invremotehostalqd3BS[[deptx[[ix]]]]=invremotehostalqd3BS[[deptx[[ix]]]]+invremotehostalqx3BS[[ix]],{ix,nx}]
Print[invremotehostalqd3BS]

```

```

(* Acumula a nivel departamental la variable invremotehostalqx3BS imputable al servicio de Circuitos Rango B
Satelital*)

Print["invremotehostald3CS"]
invremotehostald3CS=zerod;
Do[invremotehostald3CS[[deptx[[ix]]]]=invremotehostald3CS[[deptx[[ix]]]]+invremotehostalqx3CS[[ix]],{ix,nx}]
Print[invremotehostald3CS]
(* Acumula a nivel departamental la variable invremotehostalqx3CS imputable al servicio de Circuitos Rango C
Satelital*)

Print["invremotehostald3ABC2"]
invremotehostald3ABC2=zerod;
Do[invremotehostald3ABC2[[deptx[[ix]]]]=invremotehostald3ABC2[[deptx[[ix]]]]+invremotehostalqx3ABC2[[ix]],{ix,nx
}]
Print[invremotehostald3ABC2]
(* Acumula a nivel departamental la variable invremotehostalqx3ABC2 imputable al servicio de Otros Circuitos
Rango A, B y C*)

Print["invremotehostald3ABCnop"]
invremotehostald3ABCnop=zerod;
Do[invremotehostald3ABCnop[[deptx[[ix]]]]=invremotehostald3ABCnop[[deptx[[ix]]]]+invremotehostalqx3ABCnop[[ix]],
{ix,nx}]
Print[invremotehostald3ABCnop]
(* Acumula a nivel departamental la variable invremotehostalqx3ABCnop imputable al servicio de Circuitos de No
Operadores*)

Print["invremotehostitxd3"]
invremotehostitxd3=zerod;
Do[invremotehostitxd3[[deptx[[ix]]]]=invremotehostitxd3[[deptx[[ix]]]]+invremotehostitxx3[[ix]],{ix,nx}]
Print[invremotehostitxd3]
(* Acumula a nivel departamental la variable invremotehostitxx3 imputable al servicio de Circuitos de
Interconexión*)

Print["invremotehostadsld3"]
invremotehostadsld3=zerod;
Do[invremotehostadsld3[[deptx[[ix]]]]=invremotehostadsld3[[deptx[[ix]]]]+invremotehostadslx3[[ix]],{ix,nx}]
Print[invremotehostadsld3]
(* Acumula a nivel departamental la variable invremotehostadslx3 imputable al servicio de ADSL*)

Print["invremotehostTOTALd3"]
invremotehostTOTALd3=zerod;
Do[invremotehostTOTALd3[[deptx[[ix]]]]=invremotehostTOTALd3[[deptx[[ix]]]]+invremotehostTOTALx3[[ix]],{ix,nx}]
Print[invremotehostTOTALd3]
(* Acumula a nivel departamental la variable invremotehostTOTALx3 imputable a todos los servicios*)

Print["invremotehostx"]
invremotehostx=invremotehostx1+invremotehostx2+invremotehostx3;
Print[invremotehostx]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Voz *)

Print["invremotehostLocx"]
invremotehostLocx=invremotehostLocx1+invremotehostLocx2+invremotehostLocx3;
Print[invremotehostLocx]
(* Total de inversiones en Tx a nivel RH imputable al servicio Local de Voz *)

Print["invremotehostLDNx"]
invremotehostLDNx=invremotehostLDNx1+invremotehostLDNx2+invremotehostLDNx3;
Print[invremotehostLDNx]
(* Total de inversiones en Tx a nivel RH imputable al servicio LDN de Voz *)

Print["invremotehostLDIx"]
invremotehostLDIx=invremotehostLDIx1+invremotehostLDIx2+invremotehostLDIx3;
Print[invremotehostLDIx]
(* Total de inversiones en Tx a nivel RH imputable al servicio LDI de Voz *)

Print["invremotehostalqx1"]
invremotehostalqx1=invremotehostalqx1A1+invremotehostalqx2A1+invremotehostalqx3A1;
Print[invremotehostalqx1]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango A *)

Print["invremotehostalqxB1"]
invremotehostalqxB1=invremotehostalqx1B1+invremotehostalqx2B1+invremotehostalqx3B1;
Print[invremotehostalqxB1]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango B *)

Print["invremotehostalqxC1"]
invremotehostalqxC1=invremotehostalqx1C1+invremotehostalqx2C1+invremotehostalqx3C1;
Print[invremotehostalqxC1]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango C *)

Print["invremotehostalqxLL"]
invremotehostalqxLL=invremotehostalqx1LL+invremotehostalqx2LL+invremotehostalqx3LL;
Print[invremotehostalqxLL]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Local Lima *)

Print["invremotehostalqxLP"]
invremotehostalqxLP=invremotehostalqx1LP+invremotehostalqx2LP+invremotehostalqx3LP;
Print[invremotehostalqxLP]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Local Provincias *)

Print["invremotehostalqxAS"]

```



```
invremotehostalqxAS=invremotehostalqx1AS+invremotehostalqx2AS+invremotehostalqx3AS;
Print[invremotehostalqxAS]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango A Satelital*)

Print["invremotehostalqxBS"]
invremotehostalqxBS=invremotehostalqx1BS+invremotehostalqx2BS+invremotehostalqx3BS;
Print[invremotehostalqxBS]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango B Satelital*)

Print["invremotehostalqxCS"]
invremotehostalqxCS=invremotehostalqx1CS+invremotehostalqx2CS+invremotehostalqx3CS;
Print[invremotehostalqxCS]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos Rango C Satelital*)

Print["invremotehostalqxABC2"]
invremotehostalqxABC2=invremotehostalqx1ABC2+invremotehostalqx2ABC2+invremotehostalqx3ABC2;
Print[invremotehostalqxABC2]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Otros Circuitos Rango A, B y C*)

Print["invremotehostalqxABCnop"]
invremotehostalqxABCnop=invremotehostalqx1ABCnop+invremotehostalqx2ABCnop+invremotehostalqx3ABCnop;
Print[invremotehostalqxABCnop]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos de No Operadores*)

Print["invremotehostitxx"]
invremotehostitxx=invremotehostitxx1+invremotehostitxx2+invremotehostitxx3;
Print[invremotehostitxx]
(* Total de inversiones en Tx a nivel RH imputable al servicio de Circuitos de Interconexión *)

Print["invremotehostdslx"]
invremotehostadslx=invremotehostadslx1+invremotehostadslx2+invremotehostadslx3;
Print[invremotehostadslx]
(* Total de inversiones en Tx a nivel RH imputable al servicio de ADSL *)

Print["invremotehostTOTALx"]
invremotehostTOTALx=invremotehostTOTALx1+invremotehostTOTALx2+invremotehostTOTALx3;
Print[invremotehostTOTALx]
(* Total de inversiones en Tx a nivel RH imputable a todos los servicios *)

Print["invremotehostd"]
invremotehostd=zerod;
Do[invremotehostd[[deptx[[ix]]]]=invremotehostd[[deptx[[ix]]]]+invremotehostx[[ix]],{ix,nx}]
Print[invremotehostd]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Voz *)

Print["invremotehostLocd"]
invremotehostLocd=zerod;
Do[invremotehostLocd[[deptx[[ix]]]]=invremotehostLocd[[deptx[[ix]]]]+invremotehostLocx[[ix]],{ix,nx}]
Print[invremotehostLocd]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio Local de Voz *)

Print["invremotehostLDNd"]
invremotehostLDNd=zerod;
Do[invremotehostLDNd[[deptx[[ix]]]]=invremotehostLDNd[[deptx[[ix]]]]+invremotehostLDNx[[ix]],{ix,nx}]
Print[invremotehostLDNd]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio LDN de Voz *)

Print["invremotehostLDId"]
invremotehostLDId=zerod;
Do[invremotehostLDId[[deptx[[ix]]]]=invremotehostLDId[[deptx[[ix]]]]+invremotehostLDIx[[ix]],{ix,nx}]
Print[invremotehostLDId]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio LDI de Voz *)

Print["invremotehostalqdA1"]
invremotehostalqdA1=zerod;
Do[invremotehostalqdA1[[deptx[[ix]]]]=invremotehostalqdA1[[deptx[[ix]]]]+invremotehostalqxA1[[ix]],{ix,nx}]
Print[invremotehostalqdA1]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango A *)

Print["invremotehostalqdB1"]
invremotehostalqdB1=zerod;
Do[invremotehostalqdB1[[deptx[[ix]]]]=invremotehostalqdB1[[deptx[[ix]]]]+invremotehostalqxB1[[ix]],{ix,nx}]
Print[invremotehostalqdB1]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango B *)

Print["invremotehostalqdC1"]
invremotehostalqdC1=zerod;
Do[invremotehostalqdC1[[deptx[[ix]]]]=invremotehostalqdC1[[deptx[[ix]]]]+invremotehostalqxC1[[ix]],{ix,nx}]
Print[invremotehostalqdC1]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango C *)

Print["invremotehostalqdLL"]
invremotehostalqdLL=zerod;
Do[invremotehostalqdLL[[deptx[[ix]]]]=invremotehostalqdLL[[deptx[[ix]]]]+invremotehostalqxLL[[ix]],{ix,nx}]
Print[invremotehostalqdLL]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Local Lima *)

Print["invremotehostalqdLP"]
invremotehostalqdLP=zerod;
Do[invremotehostalqdLP[[deptx[[ix]]]]=invremotehostalqdLP[[deptx[[ix]]]]+invremotehostalqxLP[[ix]],{ix,nx}]
Print[invremotehostalqdLP]
```



```

(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Local Provincias *)

Print["invremotehostaldAS"]
invremotehostaldAS=zerod;
Do[invremotehostaldAS[[deptx[[ix]]]]=invremotehostaldAS[[deptx[[ix]]]]+invremotehostaldqAS[[ix]],{ix,nx}]
Print[invremotehostaldAS]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango A Satelital*)

Print["invremotehostaldBS"]
invremotehostaldBS=zerod;
Do[invremotehostaldBS[[deptx[[ix]]]]=invremotehostaldBS[[deptx[[ix]]]]+invremotehostaldqBS[[ix]],{ix,nx}]
Print[invremotehostaldBS]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango B Satelital*)

Print["invremotehostaldCS"]
invremotehostaldCS=zerod;
Do[invremotehostaldCS[[deptx[[ix]]]]=invremotehostaldCS[[deptx[[ix]]]]+invremotehostaldqCS[[ix]],{ix,nx}]
Print[invremotehostaldCS]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos Rango C Satelital*)

Print["invremotehostaldABC2"]
invremotehostaldABC2=zerod;
Do[invremotehostaldABC2[[deptx[[ix]]]]=invremotehostaldABC2[[deptx[[ix]]]]+invremotehostaldqABC2[[ix]],{ix,nx}]
Print[invremotehostaldABC2]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Otros Circuitos Rango A, B y C*)

Print["invremotehostaldABCnop"]
invremotehostaldABCnop=zerod;
Do[invremotehostaldABCnop[[deptx[[ix]]]]=invremotehostaldABCnop[[deptx[[ix]]]]+invremotehostaldqABCnop[[ix]],{ix,nx}]
Print[invremotehostaldABCnop]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos de No Operadores*)

Print["invremotehostitxd"]
invremotehostitxd=zerod;
Do[invremotehostitxd[[deptx[[ix]]]]=invremotehostitxd[[deptx[[ix]]]]+invremotehostitxx[[ix]],{ix,nx}]
Print[invremotehostitxd]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de Circuitos de Interconexión *)

Print["invremotehostadsld"]
invremotehostadsld=zerod;
Do[invremotehostadsld[[deptx[[ix]]]]=invremotehostadsld[[deptx[[ix]]]]+invremotehostadsldx[[ix]],{ix,nx}]
Print[invremotehostadsld]
(* Total de inversiones en Tx a nivel RH por departamento imputable al servicio de ADSL *)

Print["invremotehostTOTALd"]
invremotehostTOTALd=zerod;
Do[invremotehostTOTALd[[deptx[[ix]]]]=invremotehostTOTALd[[deptx[[ix]]]]+invremotehostTOTALx[[ix]],{ix,nx}]
Print[invremotehostTOTALd]
(* Total de inversiones en Tx a nivel RH por departamento imputable a todos los servicios *)

Print["invremotehost"]
invremotehost=Plus@@invremotehostd;
Print[invremotehost]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Voz *)

Print["invremotehostLoc"]
invremotehostLoc=Plus@@invremotehostLocd;
Print[invremotehostLoc]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio Local de Voz *)

Print["invremotehostLDN"]
invremotehostLDN=Plus@@invremotehostLDNd;
Print[invremotehostLDN]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio LDN de Voz *)

Print["invremotehostLDI"]
invremotehostLDI=Plus@@invremotehostLDId;
Print[invremotehostLDI]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio LDI de Voz *)

Print["invremotehostaqlA1"]
invremotehostaqlA1=Plus@@invremotehostaldA1;
Print[invremotehostaqlA1]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango A *)

Print["invremotehostaqlB1"]
invremotehostaqlB1=Plus@@invremotehostaldB1;
Print[invremotehostaqlB1]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango B *)

Print["invremotehostaqlC1"]
invremotehostaqlC1=Plus@@invremotehostaldC1;
Print[invremotehostaqlC1]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango C *)

Print["invremotehostaqlLL"]
invremotehostaqlLL=Plus@@invremotehostaldLL;
Print[invremotehostaqlLL]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Local Lima *)

```

```
Print["invremotehostaqlLP"]
invremotehostalqLP=Plus@@invremotehostalqdLP;
Print[invremotehostalqLP]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Local Provincias *)

Print["invremotehostaqlAS"]
invremotehostalqAS=Plus@@invremotehostalqdAS;
Print[invremotehostalqAS]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango A Satelital*)

Print["invremotehostaqlBS"]
invremotehostalqBS=Plus@@invremotehostalqdBS;
Print[invremotehostalqBS]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango B Satelital *)

Print["invremotehostaqlCS"]
invremotehostalqCS=Plus@@invremotehostalqdCS;
Print[invremotehostalqCS]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos Rango C Satelital*)

Print["invremotehostaqlABC2"]
invremotehostalqABC2=Plus@@invremotehostalqdABC2;
Print[invremotehostalqABC2]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Otros Circuitos Rango A, B y C*)

Print["invremotehostaqlABCnop"]
invremotehostalqABCnop=Plus@@invremotehostalqdABCnop;
Print[invremotehostalqABCnop]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos de No Operadores*)

Print["invremotehostitx"]
invremotehostitx=Plus@@invremotehostitxd;
Print[invremotehostitx]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de Circuitos de Interconexión *)

Print["invremotehostadsl"]
invremotehostadsl=Plus@@invremotehostadsl;
Print[invremotehostadsl]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de ADSL *)

Print["invremotehostTOTAL"]
invremotehostTOTAL=Plus@@invremotehostTOTAL;
Print[invremotehostTOTAL]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable a todos los servicios *)
(*****)
```

PRINT["INVERSIÓN POR SERVICIO - REMOTE HOST"]

```
Resinvremotehostx=Table[0,{jx,nx},{ix,13}];
Do[Resinvremotehostx[[ix,1]] = invremotehostx[[ix]];
  Resinvremotehostx[[ix,2]] = invremotehostalqxAl[[ix]];
  Resinvremotehostx[[ix,3]] = invremotehostalqxB1[[ix]];
  Resinvremotehostx[[ix,4]] = invremotehostalqxC1[[ix]];
  Resinvremotehostx[[ix,5]] = invremotehostalqxLL[[ix]];
  Resinvremotehostx[[ix,6]] = invremotehostalqxLP[[ix]];
  Resinvremotehostx[[ix,7]] = invremotehostalqxAS[[ix]];
  Resinvremotehostx[[ix,8]] = invremotehostalqxBS[[ix]];
  Resinvremotehostx[[ix,9]] = invremotehostalqxCS[[ix]];
  Resinvremotehostx[[ix,10]] = invremotehostalqxABC2[[ix]];
  Resinvremotehostx[[ix,11]] = invremotehostalqxABCnop[[ix]];
  Resinvremotehostx[[ix,12]] = invremotehostitxx[[ix]];
  Resinvremotehostx[[ix,13]] = invremotehostadslx[[ix]],{ix,nx}];
Print[Resinvremotehostx];
```

```
Print[Style["RESUMEN DE INVERSIONES POR ELEMENTO Y DEPARTAMENTO",
  Bold, 16, Underlined, White, Background → RGBColor[0, 1/2, 0]]];
```

```
Print[Style["A NIVEL DE CONMUTACIÓN", Bold, 14, Underlined, White, Background → Blue]];
```

```
InvRemotad = remotainvd;
InvRemotaLocd = remoteinvLocd;
InvRemotaLDNd = remotainvLDNd;
InvRemotaLDId = remotainvLDId;
Print[TableForm[
  Table[{id, deptTXTd[[id]], InvRemotad[[id]], InvRemotaLocd[[id]], InvRemotaLDNd[[id]], InvRemotaLDId[[id]]}, {id, nd}},
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Departamento", "InvRemotad", "InvRemotaLocd", "InvRemotaLDNd", "InvRemotaLDId"}}]];
(* Inversion total en Centrales remotas locales por departamento *)
```

```
InvHostd = hostsatinvd + hostnonsatinvd;
InvHostLocd = hostsatinvLocd + hostnonsatinvLocd;
InvHostLDNd = hostsatinvLDNd + hostnonsatinvLDNd;
InvHostLDId = hostsatinvLDId + hostnonsatinvLDId;
Print[
  TableForm[Table[{id, deptTXTd[[id]], InvHostd[[id]], InvHostLocd[[id]], InvHostLDNd[[id]], InvHostLDId[[id]]}, {id, nd}],
```

```

TableDirections → {Column, Row}, TableSpacing → {0, 1},
TableHeadings → {None, {"id", "Departamento", "InvHostd", "InvHostLocd", "InvHostLDNd", "InvHostLDId"}}];
(* Inversion total en Centrales cabeceras locales por departamento *)

InvTandemd = tandeminvd;
InvTandemLocd = tandeminvLocd;
InvTandemLDNd = tandeminvLDNd;
InvTandemLDId = tandeminvLDId;
Print[TableForm[
  Table[{id, deptTXTd[id], InvTandemd[id], InvTandemLocd[id], InvTandemLDNd[id], InvTandemLDId[id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Departamento", "InvTandemd", "InvTandemLocd", "InvTandemLDNd", "InvTandemLDId"}}];
(* Inversion total en Centrales tándem locales por departamento *)

InvSignal = signalfrac * (InvRemotad + InvHostd + InvTandemd);
InvSignalLocd = signalfrac * (InvRemotaLocd + InvHostLocd + InvTandemLocd);
InvSignalLDNd = signalfrac * (InvRemotaLDNd + InvHostLDNd + InvTandemLDNd);
InvSignalLDId = signalfrac * (InvRemotaLDId + InvHostLDId + InvTandemLDId);
Print[TableForm[
  Table[{id, deptTXTd[id], InvSignal[id], InvSignalLocd[id], InvSignalLDNd[id], InvSignalLDId[id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Departamento", "InvSignal", "InvSignalLocd", "InvSignalLDNd", "InvSignalLDId"}}];
(* Inversion total en señalización de todas las centrales locales por departamento *)

Print[Style["A NIVEL DE TRANSMISIÓN", Bold, 14, Underlined, White, Background → Blue]];

InvRemotaHostd = invremotehostd;
InvRemotaHostLocd = invremotehostLocd;
InvRemotaHostLDNd = invremotehostLDNd;
InvRemotaHostLDId = invremotehostLDId;
Print[TableForm[Table[{id, deptTXTd[id], InvRemotaHostd[id], InvRemotaHostLocd[id], InvRemotaHostLDNd[id],
  InvRemotaHostLDId[id]}, {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings → {None,
  {"id", "Departamento", "InvRemotaHostd", "InvRemotaHostLocd", "InvRemotaHostLDNd", "InvRemotaHostLDId"}}];
(* Inversion total en Transmisión a nivel RH por departamento imputable
al servicio de Voz *)

Print["InvRemotaHostd"];
InvRemotaHostalqdA1 = invremotehostalqdA1;
InvRemotaHostalqdB1 = invremotehostalqdB1;
InvRemotaHostalqdC1 = invremotehostalqdC1;
InvRemotaHostalqdLL = invremotehostalqdLL;
InvRemotaHostalqdLP = invremotehostalqdLP;
InvRemotaHostalqdAS = invremotehostalqdAS;
InvRemotaHostalqdBs = invremotehostalqdBs;
InvRemotaHostalqdCS = invremotehostalqdCS;
InvRemotaHostalqdABC2 = invremotehostalqdABC2;
InvRemotaHostalqdABCnop = invremotehostalqdABCnop;
InvRemotaHostitxd = invremotehostitxd;
InvRemotaHostadsld = invremotehostadsld;
InvRemotaHostTOTALd = invremotehostTOTALd;
Print[TableForm[Table[{id, deptTXTd[id], InvRemotaHostd[id], InvRemotaHostalqdA1[id],
  InvRemotaHostalqdB1[id], InvRemotaHostalqdC1[id], InvRemotaHostalqdLL[id], InvRemotaHostalqdLP[id],
  InvRemotaHostalqdAS[id], InvRemotaHostalqdBs[id], InvRemotaHostalqdCS[id], InvRemotaHostalqdABC2[id],
  InvRemotaHostalqdABCnop[id], InvRemotaHostitxd[id], InvRemotaHostadsld[id], InvRemotaHostTOTALd[id]},
{id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1},
TableHeadings → {None, {"id", "Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov",
  "Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}];
(* Inversion total en Transmisión a nivel RH por departamento imputable al cada servicio *)

InvHostTandemd = invhosttandemd;
InvHostTandemLocd = invhosttandemLocd;
InvHostTandemLDNd = invhosttandemLDNd;
InvHostTandemLDId = invhosttandemLDId;
Print[TableForm[Table[{id, deptTXTd[id], InvHostTandemd[id], InvHostTandemLocd[id], InvHostTandemLDNd[id],
  InvHostTandemLDId[id]}, {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings → {None,
  {"id", "Departamento", "InvHostTandemd", "InvHostTandemLocd", "InvHostTandemLDNd", "InvHostTandemLDId"}}];
(* Inversion total en Transmisión a nivel HT por departamento imputable al servicio de Voz *)

Print["InvHostTandemd"];
InvHostTandemalqdA1 = invhosttandemalqdA1;
InvHostTandemalqdB1 = invhosttandemalqdB1;
InvHostTandemalqdC1 = invhosttandemalqdC1;
InvHostTandemalqdLL = invhosttandemalqdLL;
InvHostTandemalqdLP = invhosttandemalqdLP;
InvHostTandemalqdAS = invhosttandemalqdAS;
InvHostTandemalqdBs = invhosttandemalqdBs;
InvHostTandemalqdCS = invhosttandemalqdCS;
InvHostTandemalqdABC2 = invhosttandemalqdABC2;
InvHostTandemalqdABCnop = invhosttandemalqdABCnop;

```

```
InvHostTandemitxd = invhosttandemitxd;
InvHostTandemadsld = invhosttandemadsld;
InvHostTandemTOTALd = invhosttandemTOTALd;
Print[TableForm[Table[{id, deptTXTd[id], InvHostTandemd[id], InvHostTandemalqdA1[id],
    InvHostTandemalqdB1[id], InvHostTandemalqdC1[id], InvHostTandemalqdLL[id], InvHostTandemalqdLP[id],
    InvHostTandemalqdA1[id], InvHostTandemalqdB1[id], InvHostTandemalqdC1[id], InvHostTandemalqdABC2[id],
    InvHostTandemalqdABCnop[id], InvHostTandemitxd[id], InvHostTandemadsld[id], InvHostTandemTOTALd[id]},
{id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1},
TableHeadings → {None, {"id", "Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov",
    "Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}]];
(* Inversion total en Transmisión a nivel HT por departamento imputable a cada servicio *)
```

CALCULO DE ANUALIZACION

```
Print[Style["FACTORES DE ANUALIZACIÓN", Bold, 14, Underlined, White, Background → Blue]];
OyMCx = networkopfrac + switchmaintfrac;
OyMTx = networkopfrac + transmaintfrac;
OyMSupp = networkopfrac + supportmaintfrac;
VuCx = Round[ $\frac{1}{\text{switchdeprfrac}}$ ];
VuTx1 = Round[ $\frac{1}{\text{transdeprfrac}}$ ];
VuTx2 = Round[ $\frac{1}{\text{fibertransdeprfrac}}$ ];
VuTx3 = Round[ $\frac{1}{\text{petransdeprfrac}}$ ];
VuSupp = Round[ $\frac{1}{\text{supportdeprfrac}}$ ];
VuEdif = Round[ $\frac{1}{\text{edifsupportdeprfrac}}$ ];
FactorSupp = OyMSupp +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuSupp}}}$ ;
FactorEdif = OyMSupp +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuEdif}}}$ ;
FactorCx = (OyMCx + overheadfrac) +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuCx}}}$  + supportinvfrac FactorSupp + edifsupportinvfrac FactorEdif;
FactorTx1 = (OyMTx + overheadfrac) +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuTx1}}}$  + supportinvfrac FactorSupp + edifsupportinvfrac FactorEdif;
FactorTx2 = (OyMTx + overheadfrac) +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuTx2}}}$  + supportinvfrac FactorSupp + edifsupportinvfrac FactorEdif;
FactorTx3 = (OyMTx + overheadfrac) +  $\frac{\text{retcap}}{1 - (1 + \text{retcap})^{-\text{VuTx3}}}$  + supportinvfrac FactorSupp + edifsupportinvfrac FactorEdif;
FactorOyMTx = (OyMTx + overheadfrac) + supportinvfrac FactorSupp + edifsupportinvfrac FactorEdif;
Print["Factores Anuales"];
Print[TableForm[{FactorSupp, FactorEdif, FactorCx, FactorTx1, FactorTx2, FactorTx3, FactorOyMTx},
    TableDirections → {Row, Column},
    TableHeadings → {"Soporte", "Edificios", "Cx", "Tx1", "Tx2", "Tx3", "OyMTx"}, Automatic]]];
```

```
Print[Style["ESTIMACIÓN DE LOS NIVELES DE GASTO POR ELEMENTO",
    Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];
```

```
Print[Style["A NIVEL CONMUTACIÓN", Bold, 14, Underlined, White, Background → Blue]];
```

```
Print["Costo Total de Cx Remota por Dpto"];
CTCxRemotad = InvRemotad * FactorCx;
CTCxRemotalocd = InvRemotalocd * FactorCx;
CTCxRemotalDnd = InvRemotalDnd * FactorCx;
CTCxRemotalDId = InvRemotalDId * FactorCx;
Print[
    TableForm[Table[{id, deptTXTd[id], CTCxRemotad[id], CTCxRemotalocd[id], CTCxRemotalDnd[id], CTCxRemotalDId[id]},
        {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
        {None, {"id", "Nombre Departamento", "CTCxRemotad", "CTCxRemotalocd", "CTCxRemotalDnd", "CTCxRemotalDId"}}]];
(* Se estima el nivel de gasto de las centrales remotas a nivel departamental *)

Print["Costo Total de Cx Host por Dpto"];
CTCxHostd = InvHostd * FactorCx;
CTCxHostlocd = InvHostlocd * FactorCx;
CTCxHostLDnd = InvHostLDnd * FactorCx;
CTCxHostLDId = InvHostLDId * FactorCx;
Print[TableForm[Table[{id, deptTXTd[id], CTCxHostd[id], CTCxHostlocd[id], CTCxHostLDnd[id], CTCxHostLDId[id]},
    {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"id", "Nombre Departamento", "CTCxHostd", "CTCxHostlocd", "CTCxHostLDnd", "CTCxHostLDId"}}]];
(* Se estima el nivel de gasto de las centrales cabeceras a nivel departamental *)
```



```

Print["Costo Total de Cx Tandem por Dpto"];
CTCxCxTandemd = InvTandemd * FactorCx;
CTCxCxTandemLocd = InvTandemLocd * FactorCx;
CTCxCxTandemLDNd = InvTandemLDNd * FactorCx;
CTCxCxTandemLDId = InvTandemLDId * FactorCx;
Print[
  TableForm[Table[{id, deptTXTd[[id]], CTCxCxTandemd[[id]], CTCxCxTandemLocd[[id]], CTCxCxTandemLDNd[[id]], CTCxCxTandemLDId[[id]]},
    {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
    {None, {"id", "Nombre Departamento", "CTCxCxTandemd", "CTCxCxTandemLocd", "CTCxCxTandemLDNd", "CTCxCxTandemLDId"}}]];
(* Se estima el nivel de gasto de las centrales Tandem a nivel departamental *)

Print["CTCxCxTandemitxd: Costo Total de Cx para Circuitos de Interconexión en Tandems por Dpto"];
CTCxCxTandemitxd = zerod * FactorCx;
Print[TableForm[Table[{id, deptTXTd[[id]], CTCxCxTandemitxd[[id]]}, {id, nd}], TableDirections → {Column, Row},
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "CTCxCxTandemitxd"}}]];
(* Se estima el nivel de gasto de Circuitos de Interconexión en las centrales Tandem a nivel departamental *)

Print["Costo Total de Cx por Dpto"];
CTCxd = CTCxRemotad + CTCxHostd + CTCxCxTandemd;
CTCxCxLocd = CTCxRemotaLocd + CTCxHostLocd + CTCxCxTandemLocd;
CTCxCxLDNd = CTCxRemotaLDNd + CTCxHostLDNd + CTCxCxTandemLDNd;
CTCxCxLDId = CTCxRemotaLDId + CTCxHostLDId + CTCxCxTandemLDId;
Print[TableForm[Table[{id, deptTXTd[[id]], CTCxd[[id]], CTCxCxLocd[[id]], CTCxCxLDNd[[id]], CTCxCxLDId[[id]]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Nombre Departamento", "CTCxd", "CTCxCxLocd", "CTCxCxLDNd", "CTCxCxLDId"}}]];
(* Gasto Total en centrales de conmutación a nivel departamental *)

Print["Costo Total de Señalización por Dpto"];
CTSxd = InvSignal * FactorCx;
CTSxCxLocd = InvSignalLocd * FactorCx;
CTSxCxLDNd = InvSignalLDNd * FactorCx;
CTSxCxLDId = InvSignalLDId * FactorCx;
Print[TableForm[Table[{id, deptTXTd[[id]], CTSxd[[id]], CTSxCxLocd[[id]], CTSxCxLDNd[[id]], CTSxCxLDId[[id]]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Nombre Departamento", "CTSxd", "CTSxCxLocd", "CTSxCxLDNd", "CTSxCxLDId"}}]];
(* Gasto Total en Señalización por departamento *)

Print[Style["A NIVEL TRANSMISIÓN", Bold, 14, Underlined, White, Background → Blue]];
Print["CTTxRemotaHostd: Costo Total de Tx RemotaHost por Dpto imputable al servicio de Voz"];
CTTxRemotaHostd = invremotehostd1 * FactorTx1 + invremotehostd2 * FactorTx2 + invremotehostd3 * FactorTx3;
CTTxRemotaHostLocd = invremotehostLocd1 * FactorTx1 + invremotehostLocd2 * FactorTx2 + invremotehostLocd3 * FactorTx3;
CTTxRemotaHostLDNd = invremotehostLDNd1 * FactorTx1 + invremotehostLDNd2 * FactorTx2 + invremotehostLDNd3 * FactorTx3;
CTTxRemotaHostLDId = invremotehostLDId1 * FactorTx1 + invremotehostLDId2 * FactorTx2 + invremotehostLDId3 * FactorTx3;
Print[TableForm[Table[{id, deptTXTd[[id]], CTTxRemotaHostd[[id]], CTTxRemotaHostLocd[[id]],
  CTTxRemotaHostLDNd[[id]], CTTxRemotaHostLDId[[id]]}, {id, nd}], TableDirections → {Column, Row},
  TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "CTTxRemotaHostd",
  "CTTxRemotaHostLocd", "CTTxRemotaHostLDNd", "CTTxRemotaHostLDId"}}]];
(* Gasto Total en Transmisión a nivel RH por departamento imputable al servicio de Voz *)

Print["Costo Total de Tx RemotaHost por Dpto"];
CTTxRemotaHostalqdA1 =
  invremotehostalqd1A1 * FactorTx1 + invremotehostalqd2A1 * FactorTx2 + invremotehostalqd3A1 * FactorTx3;
CTTxRemotaHostalqdB1 = invremotehostalqd1B1 * FactorTx1 +
  invremotehostalqd2B1 * FactorTx2 + invremotehostalqd3B1 * FactorTx3;
CTTxRemotaHostalqdc1 = invremotehostalqd1C1 * FactorTx1 + invremotehostalqd2C1 * FactorTx2 +
  invremotehostalqd3C1 * FactorTx3;
CTTxRemotaHostalqdLL = invremotehostalqd1LL * FactorTx1 + invremotehostalqd2LL * FactorTx2 +
  invremotehostalqd3LL * FactorTx3;
CTTxRemotaHostalqdLP = invremotehostalqd1LP * FactorTx1 + invremotehostalqd2LP * FactorTx2 +
  invremotehostalqd3LP * FactorTx3;
CTTxRemotaHostalqdAS = invremotehostalqd1AS * FactorTx1 + invremotehostalqd2AS * FactorTx2 +
  invremotehostalqd3AS * FactorTx3;
CTTxRemotaHostalqdBs = invremotehostalqd1BS * FactorTx1 + invremotehostalqd2BS * FactorTx2 +
  invremotehostalqd3BS * FactorTx3;
CTTxRemotaHostalqdCS = invremotehostalqd1CS * FactorTx1 + invremotehostalqd2CS * FactorTx2 +
  invremotehostalqd3CS * FactorTx3;
CTTxRemotaHostalqdABC2 = invremotehostalqd1ABC2 * FactorTx1 +
  invremotehostalqd2ABC2 * FactorTx2 + invremotehostalqd3ABC2 * FactorTx3;
CTTxRemotaHostalqdABCnop = invremotehostalqd1ABCnop * FactorTx1 +
  invremotehostalqd2ABCnop * FactorTx2 + invremotehostalqd3ABCnop * FactorTx3;
CTTxRemotaHostitxd = invremotehostitxd1 * FactorTx1 + invremotehostitxd2 * FactorTx2 + invremotehostitxd3 * FactorTx3;
CTTxRemotaHostadsld =
  invremotehostadsld1 * FactorTx1 + invremotehostadsld2 * FactorTx2 + invremotehostadsld3 * FactorTx3;
CTTxRemotaHostTOTALd = invremotehostTOTALd1 * FactorTx1 +
  invremotehostTOTALd2 * FactorTx2 + invremotehostTOTALd3 * FactorTx3;
Print[TableForm[Table[{id, deptTXTd[[id]], CTTxRemotaHostd[[id]], CTTxRemotaHostalqdA1[[id]], CTTxRemotaHostalqdB1[[id]],

```



```

CTTxRemotaHostalqdC1[id], CTTxRemotaHostalqdLL[id], CTTxRemotaHostalqdLP[id], CTTxRemotaHostalqdAS[id],
CTTxRemotaHostalqdBS[id], CTTxRemotaHostalqdCS[id], CTTxRemotaHostalqdABC2[id], CTTxRemotaHostalqdABCnop[id],
CTTxRemotaHostitxd[id], CTTxRemotaHostadsld[id], CTTxRemotaHostTOTALd[id]], {id, nd}},
TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
{None, {"id", "Nombre Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov",
"Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}]];
(* Gasto Total en Transmisión a nivel RH por departamento imputable a todos los servicios *)

Print["CTTxHostTandemd: Costo Total de Tx HostTandem por Dpto imputable al servicio de Voz"];
CTTxHostTandemd = invhosttandemd1 * FactorTx1 + invhosttandemd2 * FactorTx2 + invhosttandemd3 * FactorTx3 +
(FactorTx1 * satinvd + Extrasatcostd) + (FactorTx1 * satlimainvd + Extrasatcostd) + interdinvpermin * satmind4;
CTTxHostTandemLocd = invhosttandemLocd1 * FactorTx1 + invhosttandemLocd2 * FactorTx2 + invhosttandemLocd3 * FactorTx3 +
(FactorTx1 * satinvLocd + ExtrasatcostLocd) + (FactorTx1 * satlimainvLocd + ExtrasatcostLocd);
CTTxHostTandemLDNd = invhosttandemLDNd1 * FactorTx1 + invhosttandemLDNd2 * FactorTx2 + invhosttandemLDNd3 * FactorTx3 +
(FactorTx1 * satinvLDNd + ExtrasatcostLDNd) + (FactorTx1 * satlimainvLDNd + ExtrasatcostLDNd);
CTTxHostTandemLDId = invhosttandemLDId1 * FactorTx1 + invhosttandemLDId2 * FactorTx2 + invhosttandemLDId3 * FactorTx3 +
(FactorTx1 * satinvLDId + ExtrasatcostLDId) + (FactorTx1 * satlimainvLDId + ExtrasatcostLDId);
Print[TableForm[Table[{id, deptTXTd[id], CTTxHostTandemd[id], CTTxHostTandemLocd[id],
CTTxHostTandemLDNd[id], CTTxHostTandemLDId[id]], {id, nd}}, TableDirections → {Column, Row},
TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Nombre Departamento", "CTTxHostTandemd",
"CTTxHostTandemLocd", "CTTxHostTandemLDNd", "CTTxHostTandemLDId"}}]];
(* Gasto Total en Transmisión a nivel HT por departamento imputable al servicio de Voz. Se incluye
el gasto asociado a los equipos de los nodos que emplean Tx satelital y el asociado al gasto por los
equipo que existen en Lima. Se añade además el gasto por el alquiler del ancho de banda satelital *)

Print["Costo Total de Tx HostTandem por Dpto"]; CTTxHostTandemalqdA1 =
invhosttandemalqd1A1 * FactorTx1 + invhosttandemalqd2A1 * FactorTx2 + invhosttandemalqd3A1 * FactorTx3 +
(FactorTx1 * satinvalqdA1 + ExtrasatcostalqdA1) + (FactorTx1 * satlimainvalqdA1 + ExtrasatcostalqdA1);
CTTxHostTandemalqdB1 = invhosttandemalqd1B1 * FactorTx1 + invhosttandemalqd2B1 * FactorTx2 + invhosttandemalqd3B1 *
FactorTx3 + (FactorTx1 * satinvalqdB1 + ExtrasatcostalqdB1) + (FactorTx1 * satlimainvalqdB1 + ExtrasatcostalqdB1);
CTTxHostTandemalqdC1 = invhosttandemalqd1C1 * FactorTx1 + invhosttandemalqd2C1 * FactorTx2 + invhosttandemalqd3C1 *
FactorTx3 + (FactorTx1 * satinvalqdC1 + ExtrasatcostalqdC1) + (FactorTx1 * satlimainvalqdC1 + ExtrasatcostalqdC1);
CTTxHostTandemalqdLL = invhosttandemalqd1LL * FactorTx1 + invhosttandemalqd2LL * FactorTx2 + invhosttandemalqd3LL *
FactorTx3 + (FactorTx1 * satinvalqdLL + ExtrasatcostalqdLL) + (FactorTx1 * satlimainvalqdLL + ExtrasatcostalqdLL);
CTTxHostTandemalqdLP = invhosttandemalqd1LP * FactorTx1 + invhosttandemalqd2LP * FactorTx2 + invhosttandemalqd3LP *
FactorTx3 + (FactorTx1 * satinvalqdLP + ExtrasatcostalqdLP) + (FactorTx1 * satlimainvalqdLP + ExtrasatcostalqdLP);
CTTxHostTandemalqdAS = invhosttandemalqd1AS * FactorTx1 + invhosttandemalqd2AS * FactorTx2 + invhosttandemalqd3AS *
FactorTx3 + (FactorTx1 * satinvalqdAS + ExtrasatcostalqdAS) + (FactorTx1 * satlimainvalqdAS + ExtrasatcostalqdAS);
CTTxHostTandemalqDBS = invhosttandemalqd1BS * FactorTx1 + invhosttandemalqd2BS * FactorTx2 + invhosttandemalqd3BS *
FactorTx3 + (FactorTx1 * satinvalqDBS + ExtrasatcostalqDBS) + (FactorTx1 * satlimainvalqDBS + ExtrasatcostalqDBS);
CTTxHostTandemalqdCS = invhosttandemalqd1CS * FactorTx1 + invhosttandemalqd2CS * FactorTx2 + invhosttandemalqd3CS *
FactorTx3 + (FactorTx1 * satinvalqdCS + ExtrasatcostalqdCS) + (FactorTx1 * satlimainvalqdCS + ExtrasatcostalqdCS);
CTTxHostTandemalqdABC2 = invhosttandemalqd1ABC2 * FactorTx1 + invhosttandemalqd2ABC2 * FactorTx2 +
invhosttandemalqd3ABC2 * FactorTx3 + (FactorTx1 * satinvalqdABC2 + ExtrasatcostalqdABC2) +
(FactorTx1 * satlimainvalqdABC2 + ExtrasatcostalqdABC2);
CTTxHostTandemalqdABCnop = invhosttandemalqd1ABCnop * FactorTx1 + invhosttandemalqd2ABCnop * FactorTx2 +
invhosttandemalqd3ABCnop * FactorTx3 + (FactorTx1 * satinvalqdABCnop + ExtrasatcostalqdABCnop) +
(FactorTx1 * satlimainvalqdABCnop + ExtrasatcostalqdABCnop);
CTTxHostTandemitxd = invhosttandemitxd1 * FactorTx1 + invhosttandemitxd2 * FactorTx2 + invhosttandemitxd3 * FactorTx3 +
(FactorTx1 * satinvitxd + Extrasatcostitxd) + (FactorTx1 * satlimainvitxd + Extrasatcostitxd);
CTTxHostTandemadsld = invhosttandemadsld1 * FactorTx1 + invhosttandemadsld2 * FactorTx2 + invhosttandemadsld3 *
FactorTx3 + (FactorTx1 * satinvadsld + Extrasatcostadsld) + (FactorTx1 * satlimainvadsld + Extrasatcostadsld);
CTTxHostTandemTOTALd = invhosttandemTOTALd1 * FactorTx1 + invhosttandemTOTALd2 * FactorTx2 +
invhosttandemTOTALd3 * FactorTx3 + (FactorTx1 * satinvTOTALd + ExtrasatcostTOTALd) +
(FactorTx1 * satlimainvTOTALd + ExtrasatcostTOTALd);
Print[TableForm[Table[{id, deptTXTd[id], CTTxHostTandemd[id], CTTxHostTandemalqdA1[id], CTTxHostTandemalqdB1[id],
CTTxHostTandemalqdC1[id], CTTxHostTandemalqdLL[id], CTTxHostTandemalqdLP[id], CTTxHostTandemalqdAS[id],
CTTxHostTandemalqDBS[id], CTTxHostTandemalqdCS[id], CTTxHostTandemalqdABC2[id], CTTxHostTandemalqdABCnop[id],
CTTxHostTandemitxd[id], CTTxHostTandemadsld[id], CTTxHostTandemTOTALd[id]], {id, nd}},
TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
{None, {"id", "Nombre Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov",
"Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}]];
(* Gasto Total en Transmisión a nivel HT por departamento imputable a todos los servicios. Se incluye
el gasto asociado a los equipos de los nodos que emplean Tx satelital y el asociado al gasto por los
equipo que existen en Lima. Se añade además el gasto por el alquiler del ancho de banda satelital *)

Print["CTTxHostTandemd: Costo Total de Tx HostTandem por Dpto imputable al servicio de Voz"];
CTTxHostTandemd = CTTxRemotaHostd + CTTxHostTandemd;
CTTxLocd = CTTxRemotaHostLocd + CTTxHostTandemLocd;
CTTxLDNd = CTTxRemotaHostLDNd + CTTxHostTandemLDNd;
CTTxLDId = CTTxRemotaHostLDId + CTTxHostTandemLDId;
Print[TableForm[Table[{id, deptTXTd[id], CTTxHostTandemd[id], CTTxLocd[id], CTTxLDNd[id], CTTxLDId[id]], {id, nd}},
TableDirections → {Column, Row}, TableSpacing → {0, 1},
TableHeadings → {None, {"id", "Nombre Departamento", "CTTxHostTandemd", "CTTxLocd", "CTTxLDNd", "CTTxLDId"}}]];
(* Gasto Total en Transmisión por departamento imputable al servicio de Voz *)

Print["Costo Total de Tx por Dpto"];
CTTxalqdA1 = CTTxRemotaHostalqdA1 + CTTxHostTandemalqdA1;
CTTxalqdB1 = CTTxRemotaHostalqdB1 + CTTxHostTandemalqdB1;

```

```

CTTxalqdC1 = CTTxRemotaHostalqdC1 + CTTxHostTandemalqdC1;
CTTxalqdLL = CTTxRemotaHostalqdLL + CTTxHostTandemalqdLL;
CTTxalqdLP = CTTxRemotaHostalqdLP + CTTxHostTandemalqdLP;
CTTxalqdAS = CTTxRemotaHostalqdAS + CTTxHostTandemalqdAS;
CTTxalqdBS = CTTxRemotaHostalqdBS + CTTxHostTandemalqdBS;
CTTxalqdCS = CTTxRemotaHostalqdCS + CTTxHostTandemalqdCS;
CTTxalqdABC2 = CTTxRemotaHostalqdABC2 + CTTxHostTandemalqdABC2;
CTTxalqdABCnop = CTTxRemotaHostalqdABCnop + CTTxHostTandemalqdABCnop;
CTTxitxd = CTTxRemotaHostitxd + CTTxHostTandemitxd;
CTTxadsld = CTTxRemotaHostadsld + CTTxHostTandemadsld;
CTTxTOTALd = CTTxRemotaHostTOTALd + CTTxHostTandemTOTALd;
Print[
  TableForm[Table[{id, deptTXTd[id], CTTxd[id], CTTxalqdA1[id], CTTxalqdB1[id], CTTxalqdC1[id], CTTxalqdLL[id],
    CTTxalqdLP[id], CTTxalqdAS[id], CTTxalqdBS[id], CTTxalqdCS[id], CTTxalqdABC2[id], CTTxalqdABCnop[id],
    CTTxitxd[id], CTTxadsld[id], CTTxTOTALd[id]}, {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1},
    TableHeadings → {None, {"id", "Nombre Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima",
      "Loc Prov", "Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}]];
(* Gasto Total en Transmisión por departamento imputable a todos los servicios *)

Print["CTTx: Gasto Total en Transmisión a nivel nacional imputable al servicio de Voz "];
CTTx = Plus @@ CTTxd;
CTTxLoc = Plus @@ CTTxLocd;
CTTxLDN = Plus @@ CTTxLDNd;
CTTxLDI = Plus @@ CTTxLDId;
Print[TableForm[{CTTx, CTTxLoc, CTTxLDN, CTTxLDI},
  TableDirections → {Row, Column}, TableHeadings → {"CTTx", "CTTxLoc", "CTTxLDN", "CTTxLDI"}, None]]];
(* Gasto Total en Transmisión a nivel nacional imputable al servicio de Voz *)

Print[Style["PARA HALLAR GASTOS ÚNICOS ADSL",
  Bold, 16, Underlined, White, Background → RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["Incógnitas"];
X1;
X2;
(*La mitad del ingreso a Intradepartamental *)

InvUsu =  $\frac{X1}{3}$ ;
InvATM =  $\frac{X2}{3}$ ;

Print["FactorTx"];
FactorTx = {FactorTx1, FactorTx2, FactorTx3};
Print[
  TableForm[FactorTx, TableDirections → {Row, Column}, TableHeadings → {"FactorTx1", "FactorTx2", "FactorTx3"}]]];

Print["InvIntraDepADSLd"];
InvIntraDepADSLd = {invremotehostadsld1 + invhosttandemadsld1 + satinvadsld + satlimainvadsld,
  invhosttandemadsld2 + invremotehostadsld2, invremotehostadsld3 + invhosttandemadsld3};
Print[TableForm[Table[{id, deptTXTd[id], InvIntraDepADSLd[1, id], InvIntraDepADSLd[2, id],
  InvIntraDepADSLd[3, id]}, {id, nd}], TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Nombre Departamento", "InvIntraDepADSL1d", "InvIntraDepADSL2d", "InvIntraDepADSL3d"}}]];

Print["InvUsuPropd"];
InvUsuPropd = Table[0, {if, 3}, {id, nd}];
Do[InvUsuPropd[if, id] = InvUsu *  $\frac{\text{InvIntraDepADSLd}[if, id]}{\text{Plus @@ Plus @@ InvIntraDepADSLd}}$ , {if, 3}, {id, nd}];
Print[TableForm[Table[{id, deptTXTd[id], InvUsuPropd[1, id], InvUsuPropd[2, id], InvUsuPropd[3, id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Nombre Departamento", "InvUsuProp1d", "InvUsuProp2d", "InvUsuProp3d"}}]];

Print["InvATMPropd"];
InvATMPropd = Table[0, {if, 3}, {id, nd}];
Do[InvATMPropd[if, id] = InvATM *  $\frac{\text{InvIntraDepADSLd}[if, id]}{\text{Plus @@ Plus @@ InvIntraDepADSLd}}$ , {if, 3}, {id, nd}];
Print[TableForm[Table[{id, deptTXTd[id], InvATMPropd[1, id], InvATMPropd[2, id], InvATMPropd[3, id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Nombre Departamento", "InvATMProp1d", "InvATMProp2d", "InvATMProp3d"}}]];

Print["InvIntraDepADSLd"];
InvIntraDepADSLd = InvIntraDepADSLd - InvUsuPropd - InvATMPropd;
Print[TableForm[
  Table[{id, deptTXTd[id], InvIntraDepADSLd[1, id], InvIntraDepADSLd[2, id], InvIntraDepADSLd[3, id]}, {id, nd}],
  TableDirections → {Column, Row}, TableSpacing → {0, 1}, TableHeadings →
  {None, {"id", "Nombre Departamento", "InvIntraDepADSL1d", "InvIntraDepADSL2d", "InvIntraDepADSL3d"}}]];

Print["CosteExtraSatADSL"];
CosteExtraSatADSL = 2 * Plus @@ Extrasatcostadsld;
Print[TableForm[CosteExtraSatADSL]];

```

```
Print["CTTxadsld"]; CTTxadsld = FullSimplify[InvIntraDepADSLd[[1]] * FactorTx[[1]] +
  InvIntraDepADSLd[[2]] * FactorTx[[2]] + InvIntraDepADSLd[[3]] * FactorTx[[3]] + 2 * Extrasatcostadsld];
Print[TableForm[Table[{id, deptTXTd[[id]], CTTxadsld[[id]]}, {id, nd}], TableDirections -> {Column, Row},
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"id", "Nombre Central", "CTTxadsld"}}]]];

Print[Style["GASTOS FINALES A NIVEL NACIONAL",
  Bold, 16, Underlined, White, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];

Print["Gasto Total en Transmisión a nivel nacional"];
CTTxalqA1 = Plus @@ CTTxalqdA1;
CTTxalqB1 = Plus @@ CTTxalqdB1;
CTTxalqC1 = Plus @@ CTTxalqdC1;
CTTxalqABC2 = Plus @@ CTTxalqdABC2;
CTTxalqABCnop = Plus @@ CTTxalqdABCnop;
CTTxitx = Plus @@ CTTxitxd;
CTTxadsl = Plus @@ CTTxadsld;
CTTxTOTAL = Plus @@ CTTxTOTALd;
Print[TableForm[{CTTx, CTTxalqA1, CTTxalqB1, CTTxalqC1, CTTxalqLL, CTTxalqLP, CTTxalqAS, CTTxalqBS,
  CTTxalqCS, CTTxalqABC2, CTTxalqABCnop, CTTxitx, CTTxadsl, CTTxTOTAL}, TableDirections -> {Row, Column},
  TableHeadings -> {"Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov", "Alq A Sat",
    "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}, None}}]];
(* Gasto Total en Transmisión a nivel nacional imputable a todos los servicios *)

Print[Style["A NIVEL TOTAL", Bold, 14, Underlined, White, Background -> Blue]];
Print["CTd: Costo Total Anual de los Elementos de red por Dpto"];
CTd = CTCxRemotad + CTCxHostd + CTCxTandemd + CTSxd + CTTxRemotaHostd + CTTxHostTandemd;
CTLocd = CTCxRemotaLocd + CTCxHostLocd + CTCxTandemLocd + CTSxLocd + CTTxRemotaHostLocd + CTTxHostTandemLocd;
CTLDNd = CTCxRemotaLDNd + CTCxHostLDNd + CTCxTandemLDNd + CTSxLDNd + CTTxRemotaHostLDNd + CTTxHostTandemLDNd;
CTLDId = CTCxRemotaLDId + CTCxHostLDId + CTCxTandemLDId + CTSxLDId + CTTxRemotaHostLDId + CTTxHostTandemLDId;
Print[TableForm[Table[{id, deptTXTd[[id]], CTd[[id]], CTTxLocd[[id]], CTLDNd[[id]], CTLDId[[id]]}, {id, nd}],
  TableDirections -> {Column, Row}, TableSpacing -> {0, 1},
  TableHeadings -> {None, {"id", "Nombre Departamento", "CTd", "CTLocd", "CTLDNd", "CTLDId"}}]]];
(* Gasto Total por departamento imputable al servicio de Voz. Incluye todos los
  elementos de red: centrales remotas, centrales cabeceras,
  centrales tandem, transmisión RH, transmisión HT y señalización *)

Print["Costo Total Anual de los Elementos de red por Dpto"];
CTalqdA1 = CTTxRemotaHostalqdA1 + CTTxHostTandemalqdA1;
CTalqdB1 = CTTxRemotaHostalqdB1 + CTTxHostTandemalqdB1;
CTalqdC1 = CTTxRemotaHostalqdC1 + CTTxHostTandemalqdC1;
CTalqdLL = CTTxRemotaHostalqdLL + CTTxHostTandemalqdLL;
CTalqdLP = CTTxRemotaHostalqdLP + CTTxHostTandemalqdLP;
CTalqdAS = CTTxRemotaHostalqdAS + CTTxHostTandemalqdAS;
CTalqdBs = CTTxRemotaHostalqdBs + CTTxHostTandemalqdBs;
CTalqdCS = CTTxRemotaHostalqdCS + CTTxHostTandemalqdCS;
CTalqdABC2 = CTTxRemotaHostalqdABC2 + CTTxHostTandemalqdABC2;
CTalqdABCnop = CTTxRemotaHostalqdABCnop + CTTxHostTandemalqdABCnop;
CTitxd = (*CTCxTandemitxd+*)CTTxRemotaHostitxd + CTTxHostTandemitxd;
CTadsld = CTTxRemotaHostadsld + CTTxHostTandemadsld;
CTTOTALd =
  CTCxRemotad + CTCxHostd + CTCxTandemd + CTSxd + CTCxTandemitxd + CTTxRemotaHostTOTALd + CTTxHostTandemTOTALd;
Print[TableForm[Table[{id, deptTXTd[[id]], CTd[[id]], CTalqdA1[[id]], CTalqdB1[[id]], CTalqdC1[[id]], CTalqdLL[[id]],
  CTalqdLP[[id]], CTalqdAS[[id]], CTalqdBs[[id]], CTalqdCS[[id]], CTalqdABC2[[id]], CTalqdABCnop[[id]], CTitxd[[id]],
  CTadsld[[id]], CTTOTALd[[id]]}, {id, nd}], TableDirections -> {Column, Row}, TableSpacing -> {0, 1},
  TableHeadings -> {None, {"id", "Nombre Departamento", "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima",
    "Loc Prov", "Alq A Sat", "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}}]]];
(* Gasto Total por departamento imputable a todos los servicios. Incluye todos los
  elementos de red: centrales remotas, centrales cabeceras,
  centrales tandem, transmisión RH, transmisión HT y señalización *)

Print["CT: Costo Total a nivel nacional de Voz"];
CT = Plus @@ CTd;
CTLoc = Plus @@ CTLocd;
CTLDN = Plus @@ CTLDNd;
CTLDI = Plus @@ CTLDId;
Print[TableForm[{CT, CTTxLoc, CTLDN, CTLDI},
  TableDirections -> {Row, Column}, TableHeadings -> {"CT", "CTLoc", "CTLDN", "CTLDI"}, None}}]];
(* Gasto Total a nivel nacional imputable al servicio de Voz. Incluye todos los
  elementos de red: centrales remotas, centrales cabeceras,
  centrales tandem, transmisión RH, transmisión HT y señalización *)

Print["Costo Total a nivel nacional"];
CTalqA1 = Plus @@ CTalqdA1;
CTalqB1 = Plus @@ CTalqdB1;
CTalqC1 = Plus @@ CTalqdC1;
CTalqLL = Plus @@ CTalqdLL;
CTalqLP = Plus @@ CTalqdLP;
```

```
CTalqAS = Plus @@ CTalqdAS;
CTalqBS = Plus @@ CTalqdBs;
CTalqCS = Plus @@ CTalqdCS;
CTalqABC2 = Plus @@ CTalqdABC2;
CTalqABCnop = Plus @@ CTalqdABCnop;
CTitx = Plus @@ CTitxd;
CTadsl = Plus @@ CTadsl;
CTTOTAL = Plus @@ CTTOTALd;
Print[TableForm[{CT, CTalqA1, CTalqB1, CTalqC1, CTalqLL, CTalqLP, CTalqAS, CTalqBS,
  CTalqCS, CTalqABC2, CTalqABCnop, CTitx, CTadsl, CTTOTAL}, TableDirections -> {Row, Column},
  TableHeadings -> {{ "Voz", "Alq A", "Alq B", "Alq C", "Loc Lima", "Loc Prov", "Alq A Sat",
    "Alq B Sat", "Alq C Sat", "Alq ABC Otros", "Alq No Op", "Itx", "ADSL", "TOTAL"}, None}]];
(* Gasto Total a nivel nacional imputable a todos los servicios. Incluye todos los elementos de
red: centrales remotas, centrales cabeceras, centrales tandem, transmisión RH, transmisión HT y señalización *)
```

RESULTADOS

Resultado final para Terminación de Llamada Osiptel

```
Print[Style["RESULTADO FINAL PARA TERMINACIÓN DE LLAMADA OSIPTEL",
  Bold, 16, Underlined, White, Background -> RGBColor[0, 1/2, 0]]];
Print["Ponderación por Dpto."];
Print["Código hecho por OSIPTEL"];
TrafTotd = zerod;
Do[TrafTotd[[deptx[[ix]]]] = TrafTotd[[deptx[[ix]]]] +
  (locoutx[[ix]] + interprovoutx[[ix]] + ldx[[ix]] + recldx[[ix]] + intlrx[[ix]] + intloutx[[ix]]), {ix, nx}];
Print["TrafTotd"];
Print[TableForm[Table[{id, deptTXTd[[id]], TrafTotd[[id]]}, {id, nd}],
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"id", "Nombre Departamento", "TrafTotd"}}]];
(* Tráfico Total *)

Print["CMed: Costo Unitario por Departamento"];
CMed = zerod;
Do[CMed[[id]] = CTd[[id]] / TrafTotd[[id]], {id, nd}];
Print[TableForm[Table[{id, deptTXTd[[id]], CMed[[id]]}, {id, nd}],
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"id", "Nombre Departamento", "CMed"}}]];

Print["TrafTotNac"];
TrafTotNac = Plus @@ TrafTotd;
Print[TrafTotNac];
(* Se coloca la información ya que sólo se había considerado 11 meses *)

Print["Ponderación Nacional de Tráfico"];
PondNacd = zerod;
Do[PondNacd[[id]] = TrafTotd[[id]] / TrafTotNac, {id, nd}];
Print[TableForm[Table[{id, deptTXTd[[id]], PondNacd[[id]]}, {id, nd}],
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"id", "Nombre Departamento", "PondNacd"}}]];

Print["Ponderación Nacional de Tráfico"];
TrafTotropNac = 0;
Do[TrafTotropNac = TrafTotropNac + If[id == LIMA, 0, 1] * TrafTotd[[id]], {id, nd}]
PondropNacd = zerod;
Do[PondropNacd[[id]] = TrafTotd[[id]] * If[id == LIMA, 0, 1] / TrafTotropNac, {id, nd}];

Print["Costo de Interconexión Promedio Ponderado a nivel Nacional - Resultado de OSIPTEL"];
CMeMinNacT = CMed . PondNacd;
Print[CMeMinNacT];
```


Resultado final para Terminación de Llamada Osiptel (por Capacidad)

```

Print[Style["CALCULO DEL CARGO FIJO PERIODICO (POR CAPACIDAD)",
  Bold, 16, Underlined, White, Background → RGBColor[0, 1/2, 0]]];

(*Función Tráfico en función de cantidad de Troncales y probabilidad de bloqueo*)
Print["Definición de Función Traf"];
Off[FindRoot::lstol];
Off[FindRoot::jsing];
Traf[n_, p_] := Module[{Expr, Sol},
  If[n > $MaxRootDegree, $MaxRootDegree = n, $MaxRootDegree = 1000];
  Expr = ExpandAll[ $\sum_{j=0}^n \frac{\lambda^j}{j!}$ ];
  Sol = ReplaceAll[ $\lambda$ , FindRoot[Expr -  $\frac{\lambda^n}{n!} \frac{1}{p}$ , { $\lambda$ , n + If[n ≥ 1, Log[n], 0]}]];
  N[Sol]];

(*E0s para tramos Remote-Host y Host-Tandem *)
E0sremotehostd = Table[
  Plus @@ Table[trunkx[[ix]], {ix, Select[Range[nx], deptx[[#]] == id ∧ (remote1x[[#]] == 1 ∨ sat1x[[#]] == 1) &]}], {id, nd}];
E0shosttandemd = Table[If[id == MADREDEDIOS, CalcTrunk[
  1/60 (switchendHCLDNp[PTOMALDONADOP] + locouthCp[PTOMALDONADOP] + switchendHCLDIp[PTOMALDONADOP]), 0.01], 0] -
  If[id == LIMA, (Plus @@ Table[trunkx[[hostp[[ip]]]] - Plus @@ Table[trunkx[[ix]], {ix, Select[Range[nx],
    provx[[#]] == ip ∧ (remote1x[[#]] == 1 ∨ sat1x[[#]] == 1) &]}], {ip, Select[Range[np], deptx[[hostp[[#]]]] == LIMA ∧
    host1x[[hostp[[#]]]] == 1 ∧ (sat1x[[hostp[[#]]]] == 0 ∨ tandem1x[[hostp[[#]]]] == 1) ∧ tandem1x[[hostp[[#]]]] == 0 &}] -
    trunkx[[idinvertx[[CentralRef]]] * 2) / 2, 0] + Plus @@ Table[trunkx[[hostp[[ip]]]] -
    Plus @@ Table[trunkx[[ix]], {ix, Select[Range[nx], provx[[#]] == ip ∧ (remote1x[[#]] == 1 ∨ sat1x[[#]] == 1) &]}],
    {ip, Select[Range[np], deptx[[hostp[[#]]]] == id ∧ host1x[[hostp[[#]]]] == 1 ∧
      (sat1x[[hostp[[#]]]] == 0 ∨ tandem1x[[hostp[[#]]]] == 1) ∧ tandem1x[[hostp[[#]]]] == 0 &}]}, {id, nd}];

(*Peso tramos Remote-Host y Host-Tandem*)
Wremotehostd = E0sremotehostd / (E0sremotehostd + E0shosttandemd);
Whosttandemd = E0shosttandemd / (E0sremotehostd + E0shosttandemd);
Print[TableForm[Table[{id, deptTXTd[id], E0sremotehostd[id], E0shosttandemd[id],
  E0sremotehostd[id] + E0shosttandemd[id], Wremotehostd[id], Whosttandemd[id]}, {id, nd}], TableSpacing → {0, 1},
  TableHeadings → {None, {"id", "Departamento", "E0s R-C", "E0s C-T", "Total E0s", "w1", "w2"}}]];

Print[Style["Tramo Remota Cabecera", Bold, 14, Underlined, White, Background → Blue]];

(*Número de Remotas por Departamento*)
Remotasd = Table[Plus @@ Table[remote1x[[ix]] + sat1x[[ix]], {ix, Select[Range[nx], deptx[[#]] == id &]}], {id, nd}];

(*Circuitos por Tramo Remota Cabecera*)
trunkremotehostd = Table[If[Remotasd[id] == 0, 0, Ceiling[E0sremotehostd[id] / Remotasd[id]]], {id, nd}];

(*Erlangs Cursados por Tramo Remota Cabecera*)
erlangremotehostd = Table[Traf[trunkremotehostd[id], prob] * (1 - prob), {id, nd}];

(*Minutos Cursados por Circuito en Tramo Remota Cabecera*)
mintrunkremotehostd = Table[60 If[trunkremotehostd[id] == 0, 0, erlangremotehostd[id] / trunkremotehostd[id]], {id, nd}];
Print[TableForm[
  Table[{id, deptTXTd[id], Remotasd[id], trunkremotehostd[id], erlangremotehostd[id], mintrunkremotehostd[id]},
    {id, nd}], TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Departamento", "Remotas",
      "Circ Remota Cabecera", "Erlang Remota Cabecera", "Min Remota Cabecera por Circuito"}}]];

Print[Style["Tramo Cabecera Tandem", Bold, 14, Underlined, White, Background → Blue]];

(*Número de Cabeceras por Departamento*)
Cabecerasd = Table[Plus @@ Table[host1x[[ix]] - sat1x[[ix]], {ix, Select[Range[nx], deptx[[#]] == id &]}], {id, nd}];

(*Circuitos por Tramo Cabecera Tandem*)
trunkhosttandemd = Table[If[Cabecerasd[id] == 0, 0, Ceiling[E0shosttandemd[id] / Cabecerasd[id]]], {id, nd}];

(*Erlangs Cursados por Tramo Cabecera Tandem*)
erlanghosttandemd = Table[Traf[trunkhosttandemd[id], prob] * (1 - prob), {id, nd}];

(*Minutos Cursados por Circuito en Tramo Cabecera Tandem*)
mintrunkhosttandemd = Table[60 If[trunkhosttandemd[id] == 0, 0, erlanghosttandemd[id] / trunkhosttandemd[id]], {id, nd}];
Print[TableForm[
  Table[{id, deptTXTd[id], Cabecerasd[id], trunkhosttandemd[id], erlanghosttandemd[id], mintrunkhosttandemd[id]},
    {id, nd}], TableSpacing → {0, 1}, TableHeadings → {None, {"id", "Departamento", "Cabeceras",
      "Circ Cabecera Tandem", "Erlang Cabecera Tandem", "Min Cabecera Tandem por Circuito"}}]];

Print[Style["Totales", Bold, 14, Underlined, White, Background → Blue]];

```



```

(*Minutos Cursados Total utilizando pesos*)
mintrunkd = mintrunkremotehostd * Wremotehostd + mintrunkhosttandemd * Whosttandemd;
Print[TableForm[Table[{id, deptTXTd[id], mintrunkd[id]}, {id, nd}],
  TableSpacing -> {0, 1}, TableHeadings -> {None, {"id", "Departamento", "Min Totales por Circuito"}}]];
(*Tráfico de Hora Cargada por Departamento*)
TrafTotHCd =
  Table[Plus @@ Table[locouthCx[[ix]] + interprovouthCx[[ix]] + ldHCx[[ix]] + recldHCx[[ix]] + intlinHCx[[ix]] + intlouthCx[[ix]],
    {ix, Select[Range[nx], deptX[[#]] == id &]}], {id, nd}];
Print[TableForm[Table[{id, deptTXTd[id], TrafTotHCd[id]}, {id, nd}], TableSpacing -> {0, 1},
  TableHeadings -> {None, {"id", "Departamento", "Tráfico Total"}}]];
(*Circuitos y E1s Equivalentes*)
CirqEquivd = Ceiling[ $\frac{\text{TrafTotHCd}}{\text{mintrunkd}}$ ];
E1sEquivd = Ceiling[ $\frac{\text{Ceiling}[\frac{\text{CirqEquivd}}{30}]}{\text{fiberfill}}$ ];
Print[TableForm[Table[{id, deptTXTd[id], CirqEquivd[id], E1sEquivd[id]}, {id, nd}], TableSpacing -> {0, 1},
  TableHeadings -> {None, {"id", "Departamento", "Circuitos Equivalentes", "E1s Equivalentes"}}]];
(*Ponderación Nacional de Tráfico en función a E1s*)
PondNacE1Equivd =  $\frac{\text{E1sEquivd}}{\text{Plus @@ E1sEquivd}}$ ;
CTE1mesd =  $\frac{\text{CTd}}{\text{E1sEquivd}} \frac{1}{12}$ ;
Print[TableForm[Table[{id, deptTXTd[id], PondNacE1Equivd[id], CTE1mesd[id]}, {id, nd}], TableSpacing -> {0, 1},
  TableHeadings -> {None, {"id", "Departamento", "Ponderador de E1s Equiv", "Costo por E1 Mensual"}}]];
Print["Costo de Interconexión Promedio Ponderado a nivel Nacional por Capacidad - Resultado de OSIPTEL"];
CMeE1NacT = CTE1mesd.PondNacE1Equivd;
Print[CMeE1NacT];

```

Resultado final para Terminación de Llamada Osiptel, Con Factor de Corrección

```

Print[Style["CALCULO DEL FACTOR DE CORRECCIÓN",
  Bold, 16, Underlined, White, Background -> RGBColor[0,  $\frac{1}{2}$ , 0]]];
(*Se define kmax, como K*)
kmax = K;
(*Función de Distribución de la demanda en función del tiempo,
o periodo total T de distribución dd[0] = K, dd[T] = 0*)
dd[t_] := K -  $\frac{K}{T} t$ ;
(*Función que determina para una capacidad contratada k,
el cruce de la capacidad contratada con la función de demanda,
también es función Inversa de dd[t],  $\tau[K]=0, \tau[0]=T$ *)
 $\tau[k_] := T - \frac{T}{K} k$ ;
(*Cantidad de minutos pagados en exceso, en función de la capacidad contratada k*)
 $q[k_] := \int_0^{\tau[k]} (dd[t] - k) dt$ ;
(*Lo que se paga en función de la capacidad contratada k*)
Cix[k_] := ak * k + aq * q[k];
(*El qmax será la cantidad de minutos a pagar, cuando k=0, o sea todo se paga por tiempo*)
qmax = q[0];
(*Los cargos están dados por*)
 $ak = \frac{F}{kmax}$ ;
 $aq = \frac{F}{qmax}$ ;
(*Donde F es el pago total si todo estuviera en sólo uno de los modos*)
(*Se optimiza el pago Cix,
se deriva respecto de la capacidad contratada k y se iguala a cero el resultado será el k óptimo: kop ;
para dicho kop, existe un qop dado por qop=q[kop]*)
kop = Replace[k, Solve[ $\partial_k Cix[k] == 0$ , k][[1]]];
qop = q[kop];
(*El Factor de corrección estará dado por la relación entre el pago total F y el pago deducido como óptimo*)
Print["Factor de Corrección"];
 $\lambda = \frac{F}{kop ak + qop aq}$ ;
Print[ $\lambda$ ];
(*Cargos Corregidos*)
Print["Costo de Interconexión Promedio Ponderado a
  nivel Nacional por Tiempo y por Capacidad - Resultados de OSIPTEL Corregidos"];
CMeMinNacTCorr = CMeMinNacT *  $\lambda$ ;
CMeE1NacTCorr = Ceiling[CMeE1NacT *  $\lambda$ ];
Print[CMeMinNacTCorr];
Print[CMeE1NacTCorr];

```

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.