

MODELO QUE DETERMINA EL VALOR PORCENTUAL DE COSTOS EN LA RED ATM-ADSL

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

```
<<Miscellaneous`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["E:\\Data\\Confidencial\\Modelo Integral"];
```

CARGA DE PARÁMETROS GENERALES

Parámetros de Area Local

El modelo considera la existencia de Areas Locales. En el Perú, el Area Local está definida como la División Política de mayor jerarquía: el Departamento.

En esta sección se identifica cada Departamento y se le asigna un número correlativo único que lo identificará en todo el modelo.

Cabe mencionar que este número asignado no coincide con el Código de Departamento asignado por el Instituto Nacional de Estadística e Informática (INEI)

■ 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

■ Identificación y Codificación de Centrales por Jerarquía

Las centrales de conmutación se agrupan por su jerarquía. El modelo define las siguientes jeraquías:

- 1) Centro Nodal
- 2) Tandem Departamental
- 3) Cabecera (Provincial)
- 4) Remota

A continuación se realiza la identificación de las Centrales de Conmutación asignándoles un número.

■ 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;
IQUITOSCENROX=481;
PTOMALDONADOX=491;
ILOCENTROX=492;
CERRODEPASCOX=498;
PIURACENTROX=504;
JULIACAX=534;
TARAPOTOX=549;
TACNACENTROX=561;
TUMBESX=574;
PUCALLPACENTROX=581;
```

Identifica la Central de Referencia

```
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;
HUANCAVELICAP=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;
SAPOSAP= 91;
TINGOMARIAP=92;
TOCACHENUEVOP=93;

```
UCHIZAP=94;  
IQUITOS9DEOCTUBREP=95;
```

■ 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"

En el archivo de texto "input_TdP.txt" se encuentra gran cantidad de parametros diversos utilizados en el modelo.
Dichos parámetros deben ser leidosy 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.

■ 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 se 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=1/ d[[57]];
(* Factor de no linealidad de fibra óptica canalizada *)
factorNoLinealNoUrban=1/ d[[58]];
(* Factor de no linealidad de fibra óptica enterrada *)
factorNoLinealRadio=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 *)
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

```
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}];
(* 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, interdinverpermin, compresDCME, redundancyfactor, proteccionLD}, TableHeadings->{{"hop", "hopfibra", "urbanfill", "radiofill", "fiberfill", "ConsideraPRIM", "consideraRD", "interdinverpermin", "compresDCME", "redundancyfactor", "proteccionLD"}}]]];
```


■ Imprime Parámetros de Transmisión de Fibra

```
Print[TableForm[{TxFibraCosteCableKm, costedelafibraurbKm, TxFibraCosteEmpalme, TxFibraDistEntreEmpalmesKm, TxFibraCosteCanalizacionKm, TxFibraCosteCamara, TxFibraDistEntreCamarasKm, TxFibraCosteTrituboKm, TxFibraCosteCableKmLD, TxFibraLimites, TxFibraCosteFijo, TxFibraCosteTributario}, TableHeadings->{{"TxFibraCosteCableKm", "costedelafibraurbKm", "TxFibraCosteEmpalme", "TxFibraDistEntreEmpalmesKm", "TxFibraCosteCanalizacionKm", "TxFibraCosteCamara", "TxFibraDistEntreCamarasKm", "TxFibraCosteTrituboKm", "TxFibraCosteCableKmLD", "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, transmaintfrac, supportmaintfrac, overheadfrac, networkopfrac, retcap, satantena, sateqtx, costesatporMHz, AñadidoCxPorKm, AñadidoTrxPorKm, TxFibraCosteTerminacion, factrednd, locchannelinvpertrunk}, TableHeadings->{{"edifsupportinvfrac", "supportinvfrac", "switchdeprfrac", "transdeprfrac", "supportdeprfrac", "switchmaintfrac", "transmaintfrac", "supportmaintfrac", "overheadfrac", "networkopfrac", "retcap", "satantena", "sateqtx", "costesatporMHz", "AñadidoCxPorKm", "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[{deptx, provx, longdegx, longminx, longsecx, latdegx, latminx, latsecx, linesx, systemx, transtechx, esAnillopx, esAnillodx}, TableAlignments->Right, TableSpacing->{1, 0}, TableDirections->{Row, Column}, TableHeadings->{{"deptx", "provx", "longdegx", "longminx", "longsecx", "latdegx", "latminx", "latsecx", "linesx", "systemx", "transtechx", "esAnillopx", "esAnillodx"}, Automatic}]]
```

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[lima1d,TableDirections->{Column},TableSpacing->{0,0}]];
```

■ Imprime Vector de Departamentos asociados a cada Provincia

```
Print["deftp"];
Print[TableForm[deftp,TableDirections->{Column},TableSpacing->{0,0}]];
```

■ Imprime Conjunto de vectores Lima/Provincia

```
Print[TableForm[{limax, lima1x, ropx, rop1x},TableSpacing->{1,0},TableDirections->{Row,Column},TableHeadings->{{"limax","lima1x","ropx","rop1x"},Automatic}]];
```

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_] := SphericalDistance[{latdegx[[ix1]]+latminx[[ix1]]/60.0+latsecx[[ix1]]/3600.0,
                                          longdegx[[ix1]]+longminx[[ix1]]/60.0+longsecx[[ix1]]/3600.0},
                                          {latdegx[[ix2]]+latminx[[ix2]]/60.0+latsecx[[ix2]]/3600.0,
                                          longdegx[[ix2]]+longminx[[ix2]]/60.0+longsecx[[ix2]]/3600.0}];
```

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)

Identificación con valor True/False

```
Print["satx"]
satx=Table[False,{ix,nx}];
satx[[1]]=transtechx[[1]]==Satelite;
Do[If[deptx[[ix]]!=LIMA,satx[[ix]]=If[provx[[ix]]==provx[[ix-1]],satx[[ix-1]],transtechx[[ix]]==Satelite];],{ix,2,nx}];
Print[satx];
(* 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 *)
```

Identificación con valor 1/0

```
Print["sat1x"]
sat1x=Table[If[satx[[ix]],1,0],{ix,nx}];
Print[sat1x]
(* 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 *)
```

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

Identificación con valor True/False

```
Print["fiberx"]
fiberx=Table[False,{ix,nx}];
Do[fiberx[[ix]]=transtechx[[ix]]==Enterrado||transtechx[[ix]]==Urbana||transtechx[[ix]]==Aereo,{ix,nx}]
Print[fiberx]
(* Identifica los nodos que usan fibra como tecnología de Tx con "True" y los demás con "False" *)
```

Identificación con valor 1/0

```
Print["fiber1x"]
fiber1x=Table[If[fiberx[[ix]],1,0],{ix,nx}];
Print[fiber1x]
(* Identifica los nodos que usan fibra como tecnología de Tx con "1" y los demás con "0"*)
```

■ 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]]=CHACHAPOYASX;
tandemd[[ANCASH]]=HUARAZX;
tandemd[[APURIMAC]]=ABANCAYX;
tandemd[[AREQUIPA]]=AREQUIPACENTROX;
tandemd[[AYACUCHO]]=AYACUCHOX;
tandemd[[CAJAMARCA]]=CAJAMARCAX;
tandemd[[CUSCO]]=CUSCOCENTROX;
tandemd[[HUANCAVELICA]]=HUANCAVELICAX;
tandemd[[HUANUCO]]=HUANUCOX;
tandemd[[ICA]]=ICACENTROX;
tandemd[[JUNIN]]=HUANCAYOCENTROX;
tandemd[[LALIBERTAD]]=TRUJILLOIX;
tandemd[[LAMBAYEQUE]]=CHICLAYOCENTROX;
tandemd[[LIMA]]=TANDEMLIMAX;
tandemd[[LORETO]]=IQUITOSCENTROX;
tandemd[[MADREDEDIOS]]=PTOMALDONADOX;
tandemd[[MOQUEGUA]]=ILOCENTROX;
tandemd[[PASCO]]=CERRODEPASCOX;
tandemd[[PIURA]]=PIURACENTROX;
tandemd[[PUNO]]=JULIACAX;
tandemd[[SANMARTIN]]=TARAPOTOX;
tandemd[[TACNA]]=TACNACENTROX;
tandemd[[TUMBES]]=TUMBESX;
tandemd[[UCAYALI]]=PUCALLPACENTROX;
Print[tandemd]
(* Identifica para cada departamento cuál es la central que será considerada Tandem *)
```

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

Identificacion con código de Central Tandem

```
Print["tandemx"]
tandemx=Table[tandemd[[deptx[[ix]]]],{ix,nx}];
Print[tandemx]
(* Para cada central se especifica cuál es la Tandem de su departamento *)
```

Identificación con valor 1/0

```
Print["tandem1x"]
tandem1x=Table[If[tandemx[[ix]]==ix,1,0],{ix,nx}];
Print[tandem1x]
(* Para aquellas centrales que son Tandem las identifica con "1" y las que no son con "0" *)
```

■ Identificación de Centrales Cabeceras (variables host1x, hostx)

- Identificación con valor 1/0

```
Print["host1x"]
host1x=Table[systemx[[ix]],{ix,nx}];
Print[host1x]
(* Para aquellas centrales que son Cabeceras las identifica con "1" y las que no son con "0" *)
```

■ Identificación con valor true/false

```
Print["hostx"]
hostx=Table[If[host1x[[ix]]==1,True,False],{ix,nx}];
Print[hostx]
(* Para aquellas centrales que son Cabeceras las identifica con "True" y las que no son con "False" *)
```

■ Identificación de Centrales Host por Provincia (variable hostp)

```
Print["hostp"];
hostp=zerop;
Do[If[hostx[[ix]],hostp[[provx[[ix]]]]=ix];,{ix,nx}];
Print[hostp];
(* Identificamos el host por cada provincia *)
```

■ Identificación de Conexiones entre centrales

■ Crea Variable Auxiliar de Conexiones (variable Conexion)

```
Print["Conexion"]
Conexion=Table[ix,{ix,nx}];
Print[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[AnilloCabeceras];
(* Anillos Cabeceras, marca las cabeceras provinciales *)
```

■ Crea Variable Auxiliar de Conexion 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[ConexionProv];
(* 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,377];
    ConexionProv[[provx[[ix]]]=provx[[377]]];,{ix,nx}];
Print[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)

■ Identificación con valor 1/0

```
Print["remote1x"]
remote1x=Table[If[SystemX[[ix]]==0,1,0],{ix,nx}];
Print[remote1x]
(* Para aquellas centrales que son remotas las identifica con "1" y las que no son remotas con "0" *)
```

■ Identificación con valor True/False

```
Print["remotex"]
remotex=Table[If[remote1x[[ix]]==1,True,False],{ix,nx}];
Print[remotex]
(* Para aquellas centrales que son Remotas las identifica con "True" y las que no son Remotas con "False" *)
```

■ 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[rophostx]
(* 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. Observe 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"]
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}];
Print[allhostx]
(* 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 . Observe 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 *)

Print["disthostx"]
disthostx=Abs[Table[distance[ix,allhostx[[ix]]],{ix,nx}]];
Print[disthostx]
(* 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 *)
```

Calcula Factor de NoLinealidad para cada conexion

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

```
Print["factorNoLinealx"]
factorNoLinealx=onex;
Do[If[transtechx[[ix]]==Enterrado||transtechx[[ix]]==Aereo,
  factorNoLinealx[[ix]]=factorNoLinealNoUrban];
  If[transtechx[[ix]]==Urbana,
    factorNoLinealx[[ix]] =factorNoLinealUrban];
  If[transtechx[[ix]]==Radio,
    factorNoLinealx[[ix]] =factorNoLinealRadio];
  ,{ix, nx}];
Print[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=disthostx/factorNoLinealx;
Print[disthostx]
(* corrige la variable disthostx por el factor de no linealidad que le corresponde *)

Print["disttandemx"]
disttandemx=disttandemx/factorNoLinealx;
Print[disttandemx]
(* corrige la variable disttandemx por el factor de no linealidad que le corresponde *)

Print["repfibrax"]
repfibrax=zerop;
Do[If[host1x[[ix]]==1,
    repfibrax[[ix]]=disttandemx[[ix]];
    Conexion[[ix]]=ix;,
    repfibrax[[ix]]=disthostx[[ix]];
    Conexion[[ix]]=allhostx[[ix]];;,{ix,nx}];
Print[repfibrax]
(* 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[Conexion];
```

Calcula el Tamaño de los Anillos (Sumatoria de distancias)

■ Tamaño de Anillo a Nivel Provincial

```
PRINT["A NIVEL PROVINCIAL"]

(*Cada provincia es un anillo de Remotas-Cabeceras*)
(*Inicializacion de variables*)
TamAnillop=zerop;
CabAnillo=1;
ProvAnillo=1;
esAnillo=0;
costeTerminacionesFibrax = zerop;

Print["TamAnillop"]
Do[If[host1x[[iax]]==1,
    If[esAnillop[[iax]]==1,
        CabAnillo=iax; (*Cabecera del anillo*)
        ProvAnillo=provx[[CabAnillo]]; (*Provincia del anillo*)
        If[provx[[iax+1]]==ProvAnillo,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|esAnillop[[iax+1]]==0,
            TamAnillop[[ProvAnillo]]=TamAnillop[[ProvAnillo]]+
                If[esAnillop[[iax]]==0,0,1]*Abs[distance[iax,CabAnillo]]/factorNoLinealx[[iax]];
            repfibrax[[iax]]=repfibrax[[iax]]+Abs[distance[iax,CabAnillo]]/factorNoLinealx[[iax]];
            Conexion[[iax]]=CabAnillo;,
            TamAnillop[[ProvAnillo]]=TamAnillop[[ProvAnillo]]+Abs[distance[iax,iax+1]]/factorNoLinealx[[iax]];
            repfibrax[[iax+1]]=Abs[distance[iax,iax+1]]/factorNoLinealx[[iax]];
            Conexion[[iax]]=iax+1;];];,{iax,nx-1}];
Print[TamAnillop];
(* 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["A NIVEL DEPARTAMENTAL"]
```

```
(*Inicializacion de variables*)
```

```
TamAnillo = 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,
```

```
    TamAnillo[[DepAnillo]]=TamAnillo[[DepAnillo]]+If[esAnillodx[[AnilloCabeceras[[iax]]]]==NA,0,1]*
```

```
      Abs[distance[AnilloCabeceras[[iax]],TandemAnillo]]/factorNoLinealx[[AnilloCabeceras[[iax]]]];
```

```
    repfibrax[[AnilloCabeceras[[iax]]]]=repfibrax[[AnilloCabeceras[[iax]]]]+
```

```
      Abs[distance[AnilloCabeceras[[iax]],TandemAnillo]] /factorNoLinealx[[AnilloCabeceras[[iax]]]];
```

```
    ConexionProv[[provx[[AnilloCabeceras[[iax]]]]]=provx[[TandemAnillo]],
```

```
    TamAnillo[[DepAnillo]]=TamAnillo[[DepAnillo]]+
```

```
      Abs[distance[AnilloCabeceras[[iax]],AnilloCabeceras[[iax+1]]]/
```

```
      factorNoLinealx[[AnilloCabeceras[[iax]]]]];
```

```
repfibrax[[AnilloCabeceras[[iax+1]]]]=Abs[distance[AnilloCabeceras[[iax]],AnilloCabeceras[[iax+1]]]/
```

```
  factorNoLinealx[[AnilloCabeceras[[iax]]]]];
```

```
ConexionProv[[provx[[AnilloCabeceras[[iax]]]]]=provx[[AnilloCabeceras[[iax+1]]]]];];];,{iax,np-1}];
```

```
Print["TamAnillo"]
```

```
Print[TamAnillo]
```

```
(* 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["ELIMINACIÓN DE VALORES MUY BAJOS EN TAMAÑO DE ANILLOS"]
```

```
Print["Calculos posteriores"]
```

```
Do[If[TamAnillo[[iap]]<0.01,TamAnillo[[iap]]=0];,{iap,np}];
```

```
Do[If[TamAnillo[[iad]]<0.01,TamAnillo[[iad]]=0];,{iad,nd}];
```

```
Print["FIN Calculos posteriores"]
```

```
Print["TamAnillo"]
```

```
Print[TamAnillo]
```

```
Print["TamAnillo"]
```

```
Print[TamAnillo]
```

```
(* Las distancias menores a 0.01 kilómetros las hace cero *)
```


ESTIMACIÓN DE NÚMERO DE LINEAS

```
PRINT["ESTIMACIONES DE NÚMERO DE LÍNEAS"]
```

```
Print["linesp"]
linesp=zerop;
Do[linesp[[provx[[ix]]]]=linesp[[provx[[ix]]]]+linesx[[ix]];;{ix,nx}];
Print[linesp]
(* Número de Líneas que existe en cada provincia: Suma las líneas de los nodos que pertenecen a cada
provincia *)

Print["linesd"]
linesd=zerod;
Do[linesd[[deptx[[ix]]]]=linesd[[deptx[[ix]]]]+linesx[[ix]];;{ix,nx}];
Print[linesd]
(* Número de Líneas que existe en cada departamento: Suma las líneas de los nodos que pertenecen a cada
departamento *)

Print["lines"]
lines=Plus@@linesd;
Print[lines]
(* Total de líneas en la red *)

Print["remotelinesd"]
remotelinesd=zerod;
Do[remotelinesd[[deptx[[ix]]]]=remotelinesd[[deptx[[ix]]]]+remote1x[[ix]]*linesx[[ix]];;{ix,nx}];
Print[remotelinesd]
(* Total de líneas conectadas sólo a las centrales remotas: Información a nivel departamental*)

Print["tandemlinesd"]
tandemlinesd=zerod;
Do[tandemlinesd[[deptx[[ix]]]]=tandemlinesd[[deptx[[ix]]]]+tandem1x[[ix]]*linesx[[ix]];;{ix,nx}];
Print[tandemlinesd]
(* Total de líneas conectadas sólo a las centrales Tandem: Información a nivel departamental*)

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

Print["remotetandemlinesd"]
remotetandemlinesd=zerod;
Do[If[remotex[[ix]]&&(allhostx[[ix]]==tandemx[[ix]]),
remotetandemlinesd[[deptx[[ix]]]]=remotetandemlinesd[[deptx[[ix]]]]+linesx[[ix]];;{ix,nx}];
Print[remotetandemlinesd]
(* 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 *)

Print["satlinesx"]
satlinesx=linesx*sat1x;
Print[satlinesx]
(* Se especifica el número de líneas de cada una de las centrales que usan tecnología de Tx satelital *)

Print["satlinesd"]
satlinesd=zerod;
Do[satlinesd[[deptx[[ix]]]]=satlinesd[[deptx[[ix]]]]+satlinesx[[ix]];;{ix,nx}];
Print[satlinesd]
(* Se totaliza a nivel departamental el número de líneas que usan tecnología de Tx satelital *)

Print["satlinesp"]
satlinesp=zerop;
Do[satlinesp[[provx[[ix]]]]=satlinesp[[provx[[ix]]]]+satlinesx[[ix]];;{ix,nx}];
Print[satlinesp]
(* Se totaliza a nivel provincial el número de líneas que usan tecnología de Tx satelital *)

Print["interprovfracx"]
interprovfracx=zerox;
Do[interprovfracx[[ix]]=1-(linesp[[provx[[ix]]]]/linesd[[deptx[[ix]]]])^(1/locb);;{ix,nx}];
Print[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["TRATAMIENTO DEL TRÁFICO EN LA HORA CARGADA"]

```
Print["Trafico en Hora Cargada"];
TraficoRealHCx=ReadList["input_Trafico_HC.txt",{Real,Real,Real,Real,Real,Real}];
(*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[[ix]][[1]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico Intradepartamental de entrada. Le añade los minutos no tarificados *)
intradepouthC1x=Table[TraficoRealHCx[[ix]][[2]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico Intradepartamental de salida. Le añade los minutos no tarificados *)
interprovinHC1x=intradepinHC1x*interprovfracx;
(* Se calcula el Trafico Interprovincial de entrada *)
interprovouthC1x=intradepouthC1x*interprovfracx;
(* Se calcula el Trafico Interprovincial de salida *)
locinHC1x=intradepinHC1x*(onex-interprovfracx);
(* Se calcula el Trafico Intraprovincial de entrada*)
locouthC1x=intradepouthC1x*(onex-interprovfracx);
(* Se calcula el Trafico Intraprovincial de salida *)
recldnacHC1x=Table[TraficoRealHCx[[ix]][[3]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico LDN ENTRANTE. Le añade los minutos no tarificados *)
ldHC1x=Table[TraficoRealHCx[[ix]][[4]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico LDN SALIENTE. Le añade los minutos no tarificados *)
intlinHC1x=Table[TraficoRealHCx[[ix]][[5]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico LDI ENTRANTE. Le añade los minutos no tarificados *)
intloutHC1x=Table[TraficoRealHCx[[ix]][[6]],{ix,nx}*(1+trafAdicNoTarif);
(* Lectura del Tráfico LDI SALIENTE. Le añade los minutos no tarificados *)
ldintradHC1x=interprovouthC1x+interprovinHC1x;
(* Total Trafico Interprovincial: Interprovincial Entrante + Interprovincial Saliente *)
reclldHC1x=recldnacHC1x;
(* Igual que recldnacHC1x: Tráfico LDN ENTRANTE *)
locHC1x=(locinHC1x+locouthC1x)/2;
(* Tráfico Intraprovincial Promedio: (Intraprovincial Entrante + Saliente)/2 *)
intraHC1x=Table[If[linesp[[provx[[ix]]]]==
0,0,locHC1x[[ix]]*(linesx[[ix]]/linesp[[provx[[ix]]]])^(1/locb)],{ix,nx}];
(* Estimación del Tráfico Intracentral *)
Print[TableForm[{intradepinHC1x,intradepouthC1x,interprovinHC1x,interprovouthC1x,locinHC1x,locouthC1x,recldnacHC1x,ldHC1x,intlinHC1x,intloutHC1x,ldintradHC1x,reclldHC1x,locHC1x,intraHC1x},TableDirections->{Row,Column},TableAlignments->Right,TableSpacing->{1,0},TableHeadings->{{"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"},Automatic}]];
```

PRINT["HORA CARGADA EN ERLANGS"]

```

Print["locinHCx"]
locinHCx=locinHC1x/60;
Print[locinHCx]
(* Tráfico Intraprovincial Entrante en HC en minutos *)

Print["locinHCp"]
locinHCp=zerop;
Do[locinHCp[[provx[[ix]]]]=locinHCp[[provx[[ix]]]]+locinHCx[[ix]];;{ix,nx}];
Print[locinHCp]
(* Acumula por provincia el Tráfico Intraprovincial Entrante en HC en minutos *)

Print["locouthCx"]
locouthCx=locouthHC1x/60;
Print[locouthCx]
(* Tráfico Intraprovincial Saliente en HC en minutos *)

Print["locouthCp"]
locouthCp=zerop;
Do[locouthCp[[provx[[ix]]]]=locouthCp[[provx[[ix]]]]+locouthCx[[ix]];;{ix,nx}];
Print[locouthCp]
(* Acumula por provincia el Tráfico Intraprovincial Saliente en HC en minutos *)

Print["ldHCx"]
ldHCx=ldHC1x/60;
Print[ldHCx]
(* Lectura del Tráfico LDN SALIENTE en HC en minutos *)

Print["reclldHCx"]
reclldHCx=reclldHC1x/60;
Print[reclldHCx]
(* Igual que recldnachHC1x: Tráfico LDN ENTRANTE en HC en minutos *)

Print["intlouthCx"]
intlouthCx=intlouthHC1x/60;
Print[intlouthCx]
(* Lectura del Tráfico LDI SALIENTE en HC en minutos *)

Print["intlinHCx"]
intlinHCx=intlinHC1x/60;
Print[intlinHCx]
(* Lectura del Tráfico LDI ENTRANTE en HC en minutos *)

Print["ldintradHCx"]
ldintradHCx=ldintradHC1x/60;
Print[ldintradHCx]
(* Total Trafico Interprovincial en HC en minutos: Interprovincial Entrante + Interprovincial Saliente *)

Print["interprovouthCx"]
interprovouthCx=interprovouthHC1x/60;
Print[interprovouthCx]
(* Tráfico Interprovincial Saliente en la HC en minutos *)

Print["locHCx"]
locHCx=locHC1x/60;
Print[locHCx]
(* Tráfico Intraprovincial Promedio en Hc en minutos: (Intraprovincial Entrante + Saliente)/2 *)

Print["intraHCx"]
intraHCx=intraHC1x/60;
Print[intraHCx]
(* Tráfico Intranodo en minutos de HC en minutos *)

Print["intraHCp"]
intraHCp=zerop;
Do[intraHCp[[provx[[ix]]]]=intraHCp[[provx[[ix]]]]+intraHCx[[ix]];;{ix,nx}];
Print[intraHCp]
(* Acumula por provincia el Tráfico Intranodo en minutos de HC en minutos *)

```

ESTIMACION DE RUTAS DIRECTAS

PRINT["ESTIMACIONES DE RUTAS DIRECTAS"]

```

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["ESTIMACIONES A NIVEL LOCAL"]

```

(*Trafico de salida*)
locoutRDx=zerox;
restalocoutHCx=zerox;

Do[restalocoutHCx[[RD[[i,1]]]]=restalocoutHCx[[RD[[i,1]]]]-interprovoutHCx[[RD[[i,1]]]]*RD[[i,3]];,{i,nrd}];

Print["restalocoutHCx"]
Print[restalocoutHCx]
(* 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 *)

Print["ldintradHCx"]
ldintradHCx=ldintradHCx+restalocoutHCx;
Print[ldintradHCx]
(* al total del tráfico interprovincial en la HC (entrante+saliente) le resta el tráfico de las rutas directas por tráfico saliente *)

(*Trafico de entrada*)
locinRDx=zerox;
restalocinHCx=zerox;
restalocinx=zerox;

Do[ locinRDx[[RD[[i,2]]]]=locinRDx[[RD[[i,2]]]]+interprovoutHCx[[RD[[i,1]]]]*RD[[i,3]]/60;
  restalocinHCx[[RD[[i,2]]]]=restalocinHCx[[RD[[i,2]]]]-interprovoutHCx[[RD[[i,1]]]]*RD[[i,3]];,{i,nrd}];

Print["locinRDx"]
Print[locinRDx]
(* 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 *)

Print["restalocinHCx"]
Print[restalocinHCx]
(* 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 *)

Print["ldintradHCx actualizado"]
ldintradHCx=ldintradHCx+restalocinHCx;
Print[ldintradHCx]
(* al total del tráfico interprovincial en minutos de HC (entrante+saliente) se le resta ahora el tráfico de las rutas directas por tráfico entrante*)

Print["loadRDx"]
loadRDx=locinRDx;
Print[loadRDx]
(* 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["ESTIMACIONES A NIVEL LDN"]

```

Print["reclimaldHCx"]
reclimaldHCx=reclldHCx lima1x;
Print[reclimaldHCx]
(* Tráfico LDN ENTRANTE en HC anual en los nodos de Lima *)

Print["recropinterldHCx"]
recropinterldHCx=reclldHCx rop1x;
Print[recropinterldHCx]
(* Tráfico LDN ENTRANTE en HC anual en los nodos del resto del Perú *)

```

PRINT["RESUMEN DEL TRÁFICO CONMUTABLE"]

PRINT["HORA CARGADA"]

```
Print["switchendHCLocx"]
switchendHCLocx=locinHCx+locoutHCx+ldintradHCx-intraHCx;
Print[switchendHCLocx]
(* 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 *)

Print["switchendHCLDNx"]
switchendHCLDNx=ldHCx+reclimaldHCx+recropinterldHCx;
Print[switchendHCLDNx]
(* 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 tráficos LDN Entrante en los nodos de Lima y LDN Entrante en los nodos de Provincias ya
están descontados del tráfico de las Rutas Directas *)

Print["switchendHCLDIx"]
switchendHCLDIx=intloutHCx+intlínHCx;
Print[switchendHCLDIx]
(* Tráfico LDI que conmuta en minutos de HC, se suman entonces los tráficos: LDI Saliente + LDI Entrante *)

Print["switchendHCx"]
switchendHCx=switchendHCLocx+switchendHCLDNx+switchendHCLDIx;
Print[switchendHCx]
(* Tráfico total que conmuta en minutos de HC, se suman entonces los tráficos:
switchendHCLocx+switchendHCLDNx+switchendHCLDIx *)

Print["switchendHCLocp"]
switchendHCLocp=zerop;
Do[switchendHCLocp[[provx[[ix]]]]=switchendHCLocp[[provx[[ix]]]]+switchendHCLocx[[ix]];;{ix,nx}];
Print[switchendHCLocp]
(* Se acumula a nivel provincial el trafico Local que conmuta en minutos de HC *)

Print["switchendHCLDNp"]
switchendHCLDNp=zerop;
Do[switchendHCLDNp[[provx[[ix]]]]=switchendHCLDNp[[provx[[ix]]]]+switchendHCLDNx[[ix]];;{ix,nx}];
Print[switchendHCLDNp]
(* Se acumula a nivel provincial el trafico LDN que conmuta en minutos de HC *)

Print["switchendHCLDIp"]
switchendHCLDIp=zerop;
Do[switchendHCLDIp[[provx[[ix]]]]=switchendHCLDIp[[provx[[ix]]]]+switchendHCLDIx[[ix]];;{ix,nx}];
Print[switchendHCLDIp]
(* Se acumula a nivel provincial el trafico LDI que conmuta en minutos de HC *)

Print["switchendHCp"];
switchendHCp=switchendHCLocp+switchendHCLDNp+switchendHCLDIp;
Print[switchendHCp];
(* Se acumula a nivel provincial el trafico total que conmuta en minutos de HC *)
```

PRINT["IDENTIFICACIÓN DE CARGAS EN ERLANGS"]

PRINT["CARGAS SATELITALES"]

PRINT["HORA CARGADA"]

```
Print["satminHCLocx"]
satminHCLocx=switchendHCLocx sat1x;
Print[satminHCLocx];

Print["satminHCLDNx"]
satminHCLDNx=switchendHCLDNx sat1x;
Print[satminHCLDNx];

Print["satminHCLDIx"]
satminHCLDIx=switchendHCLDIx sat1x;
Print[satminHCLDIx];

Print["satldintradHCx"]
satldintradHCx=ldintradHCx sat1x;
Print[satldintradHCx];

Print["satminHCx"]
satminHCx=(satminHCLocx+satminHCLDNx+satminHCLDIx)*sat1x;
Print[satminHCx]
(* 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 *)

Print["loadsatLocx"]
```



```

loadsatLocx=satminHCLocx/60;
Print[loadsatLocx]

Print["loadsatLDNx"]
loadsatLDNx=satminHCLDNx/60;
Print[loadsatLDNx]

Print["loadsatLDIx"]
loadsatLDIx=satminHCLDIx/60;
Print[loadsatLDIx]

Print["loadsatx"]
loadsatx=satminHCx/60;
Print[loadsatx]
(* Se divide el tráfico considerado para el dimensionamiento satelital entre 60 para expresarlo en Erlangs *)

(* Tablas Ponderadoras *)
satLocx=Table[If[loadsatx[[ix]]>0, loadsatLocx[[ix]]/loadsatx[[ix]],0],{ix,nx}];
satLDNx=Table[If[loadsatx[[ix]]>0, loadsatLDNx[[ix]]/loadsatx[[ix]],0],{ix,nx}];
satLDIx=Table[If[loadsatx[[ix]]>0, loadsatLDIx[[ix]]/loadsatx[[ix]],0],{ix,nx}];

PRINT["TRAMO REMOTA-CABECERA"]

Print["ropminremotehostHCLocx"]
ropminremotehostHCLocx=Table[If[remotex[[ix]]&&ropx[[ix]], switchendHCLocx[[ix]]-intraHCx[[ix]],0],{ix,nx}];
Print[ropminremotehostHCLocx];

Print["ropminremotehostHCLDNx"]
ropminremotehostHCLDNx=Table[If[remotex[[ix]]&&ropx[[ix]], switchendHCLDNx[[ix]],0],{ix,nx}];
Print[ropminremotehostHCLDNx];

Print["ropminremotehostHCLDIx"]
ropminremotehostHCLDIx=Table[If[remotex[[ix]]&&ropx[[ix]], switchendHCLDIx[[ix]],0],{ix,nx}];
Print[ropminremotehostHCLDIx];

Print["ropminremotehostHCx"]
ropminremotehostHCx=ropminremotehostHCLocx+ropminremotehostHCLDNx+ropminremotehostHCLDIx;
Print[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 *)

Print["limaminremotehostHCLocx"]
limaminremotehostHCLocx=(switchendHCLocx-intraHCx)*(lima1x*remote1x);
Print[limaminremotehostHCLocx]

Print["limaminremotehostHCLDNx"]
limaminremotehostHCLDNx=(switchendHCLDNx)*(lima1x*remote1x);
Print[limaminremotehostHCLDNx]

Print["limaminremotehostHCLDIx"]
limaminremotehostHCLDIx=(switchendHCLDIx)*(lima1x*remote1x);
Print[limaminremotehostHCLDIx]

Print["limaminremotehostHCx"]
limaminremotehostHCx=limaminremotehostHCLocx+limaminremotehostHCLDNx+limaminremotehostHCLDIx;
Print[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 *)

Print["minremotehostHCLocx"]
minremotehostHCLocx=rop1x*ropminremotehostHCLocx+lima1x*limaminremotehostHCLocx;
Print[minremotehostHCLocx]

Print["minremotehostHCLDNx"]
minremotehostHCLDNx=rop1x*ropminremotehostHCLDNx+lima1x*limaminremotehostHCLDNx;
Print[minremotehostHCLDNx]

Print["minremotehostHCLDIx"]
minremotehostHCLDIx=rop1x*ropminremotehostHCLDIx+lima1x*limaminremotehostHCLDIx;
Print[minremotehostHCLDIx]

Print["minremotehostHCx"]
minremotehostHCx=rop1x*ropminremotehostHCx+lima1x*limaminremotehostHCx;
Print[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 *)

Print["loadremotehostLocx"]
loadremotehostLocx=minremotehostHCLocx /60;
Print[loadremotehostLocx];
(* Para todas las remotas se divide el tráfico conmutable Local entre 60 para identificar la carga
correspondiente en Erlangs en el nivel R-H *)

Print["loadremotehostLDNx"]
loadremotehostLDNx=minremotehostHCLDNx /60;
Print[loadremotehostLDNx];
(* Para todas las remotas se divide el tráfico conmutable LDN entre 60 para identificar la carga
correspondiente en Erlangs en el nivel R-H *)

Print["loadremotehostLDIx"]

```



```

loadremotehostLDIx=minremotehostHCLDIx /60;
Print[loadremotehostLDIx];
(* Para todas las remotas se divide el tráfico conmutable LDI entre 60 para identificar la carga
correspondiente en Erlangs en el nivel R-H *)

Print["loadremotehostx"]
loadremotehostx=minremotehostHCx /60;
Print[loadremotehostx];
(* Para todas las remotas se divide el tráfico conmutable entre 60 para identificar la carga correspondiente
en Erlangs en el nivel R-H *)

PRINT["TRAMO CABECERA-TANDEM"]

Print["ropminhosttandemHCLocx"]
ropminhosttandemHCLocx=Table[If[hostx[[ix]]&&ropx[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLocp[[
provx[[ix]]]]-(locinHCp[[provx[[ix]]]]+locoutHCp[[provx[[ix]]]]-intraHCp[[provx[[ix]]]]),0},{ix,nx}];
Print[ropminhosttandemHCLocx]

Print["ropminhosttandemHCLDNx"]
ropminhosttandemHCLDNx=Table[If[hostx[[ix]]&&ropx[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLDNp[[
provx[[ix]]]],0},{ix,nx}];
Print[ropminhosttandemHCLDNx]

Print["ropminhosttandemHCLDIx"]
ropminhosttandemHCLDIx=Table[If[hostx[[ix]]&&ropx[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLDIp[[
provx[[ix]]]],0},{ix,nx}];
Print[ropminhosttandemHCLDIx]

Print["ropminhosttandemHCx"]
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},{ix,nx}];
Print[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 *)

Print["limaminhosttandemHCLocx"]
limaminhosttandemHCLocx=Table[If[hostx[[ix]]&&limax[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLocp[[
provx[[ix]]]]-(locinHCp[[provx[[ix]]]]+locoutHCp[[provx[[ix]]]]-intraHCp[[provx[[ix]]]]),0},{ix,nx}];
Print[limaminhosttandemHCLocx]

Print["limaminhosttandemHCLDNx"]
limaminhosttandemHCLDNx=Table[If[hostx[[ix]]&&limax[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLDNp[[
provx[[ix]]]],0},{ix,nx}];
Print[limaminhosttandemHCLDNx]

Print["limaminhosttandemHCLDIx"]
limaminhosttandemHCLDIx=Table[If[hostx[[ix]]&&limax[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCLDIp[[
provx[[ix]]]],0},{ix,nx}];
Print[limaminhosttandemHCLDIx]

Print["limaminhosttandemHCx"]
limaminhosttandemHCx=Table[If[hostx[[ix]]&&limax[[ix]]&&(tandemx[[ix]]!=ix)&&(!satx[[ix]]),switchendHCp[[prov
x[[ix]]]]-(locinHCp[[provx[[ix]]]]+locoutHCp[[provx[[ix]]]]-intraHCp[[provx[[ix]]]]),0},{ix,nx}];
Print[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 *)

Print["minhosttandemHCLocx"]
minhosttandemHCLocx=rop1x*ropminhosttandemHCLocx+lima1x*limaminhosttandemHCLocx;
Print[minhosttandemHCLocx]

Print["minhosttandemHCLDNx"]
minhosttandemHCLDNx=rop1x*ropminhosttandemHCLDNx+lima1x*limaminhosttandemHCLDNx;
Print[minhosttandemHCLDNx]

Print["minhosttandemHCLDIx"]
minhosttandemHCLDIx=rop1x*ropminhosttandemHCLDIx+lima1x*limaminhosttandemHCLDIx;
Print[minhosttandemHCLDIx]

Print["minhosttandemHCx"]
minhosttandemHCx=rop1x*ropminhosttandemHCx+lima1x*limaminhosttandemHCx;
Print[minhosttandemHCx]
(*Sumando los dos resultados previos (Lima+Provincias) obtenemos el tráfico total que es conmutable a nivel
H-T*)

Print["loadhosttandemLocx"]
loadhosttandemLocx=minhosttandemHCLocx/60; (*Para convertir los minutos a Erlangs*)
Print[loadhosttandemLocx]

Print["loadhosttandemLDNx"]
loadhosttandemLDNx=minhosttandemHCLDNx/60; (*Para convertir los minutos a Erlangs*)
Print[loadhosttandemLDNx]

Print["loadhosttandemLDIx"]

```



```
loadhosttandemLDIx=minhosttandemHCLDIdx/60; (*Para convertir los minutos a Erlangs*)
Print[loadhosttandemLDIx]

Print["loadhosttandemx"]
loadhosttandemx=minhosttandemHCx/60; (*Para convertir los minutos a Erlangs*)
Print[loadhosttandemx]
(* 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["ESTIMACIÓN DEL NÚMERO DE TRONCALES Y DE E1s DE VOZ"]

```
(* Formula de Erlang B en General *)
(* erlangb[λ_,n_]:=((λ^n)*(n!))/((Sum[λ^j/(j!),{j,0,n}]]); *)
(* Formula para hallar el número de troncales, basado en la fórmula de Erlang B *)
Print["Definición de la función CalcTrunk"]
CalcTrunk[λ_,p_]:=Module[{n,S},n=0;S=1;If[(λ>0)&&(p<1),While[(1/S)>p,n=n+1;S=S*n/λ+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["A NIVEL SATELITAL"]

```
Print["trunksatx"]
trunksatx=zerox;
Do[trunksatx[[ix]]=CalcTrunk[loadsatx[[ix]],prob];,{ix,nx}]
Print[trunksatx]
(* Se estima el número de troncales a nivel Satelital sobre la base de las cargas en Erlangs estimadas en la sección anterior *)

Print["e1satx"]
e1satx=zerox;
Do[e1satx[[ix]]=Ceiling[Ceiling[trunksatx[[ix]]/30]],{ix,nx}]
Print[e1satx]
(* 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["A NIVEL DE RUTAS DIRECTAS"]

```
Print["trunkRDx"]
trunkRDx=zerox;
Do[trunkRDx[[ix]]=CalcTrunk[loadRDx[[ix]],prob];,{ix,nx}]
Print[trunkRDx]
(* 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 *)

Print["e1RDx"]
e1RDx=Ceiling[Ceiling[trunkRDx/30/fiberfill]];
Print[e1RDx]
(* 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 *)
```

PRINT["A NIVEL REMOTA-CABECERAS"]

```
Print["trunkremotehostx"]
trunkremotehostx=zerox;
Do[trunkremotehostx[[ix]]=CalcTrunk[loadremotehostx[[ix]],prob];,{ix,nx}]
Print[trunkremotehostx]
(* Se estima el número de troncales a nivel RH sobre la base de las cargas en Erlangs estimadas en la sección anterior *)

Print["e1remotehostx"]
e1remotehostx=zerox;
Do[If[fiberx[[ix]],e1remotehostx[[ix]]=Ceiling[Ceiling[trunkremotehostx[[ix]]/30/fiberfill]];,
If[satx[[ix]],e1remotehostx[[ix]]=Ceiling[Ceiling[trunkremotehostx[[ix]]/30]];,
e1remotehostx[[ix]]=Ceiling[Ceiling[trunkremotehostx[[ix]]/30/radiofill]]];];,{ix,nx}];
Print[e1remotehostx]
(* A partir del número de troncales estimados en el tramo RH se estima el núemro 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;
```

PRINT["A NIVEL CABECERAS-TANDEM"]

```
Print["trunkhosttandemx"]
trunkhosttandemx=zerox;
Do[trunkhosttandemx[[ix]]=CalcTrunk[loadhosttandemx[[ix]],prob];,{ix,nx}]
Print[trunkhosttandemx]
(* Se estima el número de troncales a nivel HT sobre la base de las cargas en Erlangs estimadas en la
```

```

sección anterior *)

Print["e1hosttandemx"]
e1hosttandemx=zeror;
Do[If[fiberx[[ix]],e1hosttandemx[[ix]]=Ceiling[Ceiling[trunkhosttandemx[[ix]]/30/fiberfill]];;,
    If[satx[[ix]],e1hosttandemx[[ix]]=Ceiling[Ceiling[trunkhosttandemx[[ix]]/30]];;,
    e1hosttandemx[[ix]]=Ceiling[Ceiling[trunkhosttandemx[[ix]]/30/radiofill]]];];,{ix,nx}];
Print[e1hosttandemx]
(* 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@@e1remotex;

Print["trunkx"]
trunkx=remotex*trunkremotex+hostx*(trunkhosttandemx+trunksatx);
Do[If[remotex[[ix]],trunkx[[allhostx[[ix]]]]=trunkx[[allhostx[[ix]]]+trunkremotex[[ix]]];,{ix,nx}];
Do[If[hostx[[ix]]&&(tandemx[[ix]]!=ix),
    trunkx[[tandemx[[ix]]]]=trunkx[[tandemx[[ix]]]+trunkhosttandemx[[ix]]];,{ix,nx}];
Print[trunkx]
(* Identificación del total de troncales asociadas a cada central. Para las centrales remotas es aplicable
el total trunkremotex, mientras que para las cabeceras es aplicable el total trunkhosttandemx +
trunksatx. A las centrales cabeceras les agrega las troncales de sus remotas asociadas (los trunkremotex
de sus remotas). A las centrales Tandem le suma además las troncales de sus cabeceras asociadas (los
trunkhosttandemx de sus cabeceras asociadas *)

```

PRINT["ESTIMACIÓN DE CARGAS EN E1S ASOCIADAS AL SERVICIO ADSL"]

PRINT["CIRCUITOS ADSL"]

```

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[MatrixForm[demandasADSL1,TableDirections->{Column},TableHeadings->{Automatic,{"Departamento","Nombre","Pop","2000","600","400","200","100","Plus"}}]];
(*Importación de la demanda de cada año*)

(*Se utiliza un input para la equivalencia entre Localidad ADSL y Central en la red Fija*)
Print["equivADSL"];
OpenRead["input_equiv_ADSL.txt"];
equivADSL=N[ReadList["input_equiv_ADSL.txt",{Number,Number}]];
Close["input_equiv_ADSL.txt"];
Print[equivADSL];

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

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

CxATMLimax=zeror;
(* lista de nodos ATM en Lima*)
OpenRead["NodosATMLima_6Pops.txt"];
(*Instruccion escondida. En texto blanco*)
CxATMLimax=ReadList["NodosATMLima_6Pops.txt",Real];

Print["Localidades"];
Localidades=Length[demandasADSL1];
Print[Localidades];
(* Todas las Localidades con servicio ADSL *)

Print["sumademandasADSL"];
sumademandasADSL=Table[demandasADSL1[[i,4]]+demandasADSL1[[i,5]]+demandasADSL1[[i,6]]+demandasADSL1[[i,7]]+demandasADSL1[[i,8]]+demandasADSL1[[i,9]],{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]] = demandasADSL1[[ix,1]] + demandasADSL1[[auxLoc,4]];
    demandasADSL1x[[ix,2]] = demandasADSL1[[ix,2]] + demandasADSL1[[auxLoc,5]];
    demandasADSL1x[[ix,3]] = demandasADSL1[[ix,3]] + demandasADSL1[[auxLoc,6]];

```

```

demandasADSL1x[[ix,4]] = demandasADSL1x[[ix,4]] + demandasADSL1[[auxLoc,7]];
demandasADSL1x[[ix,5]] = demandasADSL1x[[ix,5]] + demandasADSL1[[auxLoc,8]];
demandasADSL1x[[ix,6]] = demandasADSL1x[[ix,6]] + demandasADSL1[[auxLoc,9]];];,{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,4]],{i,Localidades}];
Print[demandaADSL2000];

Print["demandaADSL2000Nuc"];
demandaADSL2000Nuc=Table[demandasADSL1Nuc[[i,1]],{i,6}];
Print[demandaADSL2000Nuc];

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

Print["CentralesConExtensores"];
CentralesConExtensores=Ceiling[demandaADSL2000/40];
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 añadio instruccion de suma de comprobacion de las lineas que requieren adecuacion
utilizando extensores*)

PRINT["CARGA DE VELOCIDAD"];

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*)

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*)

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*)

```

```

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*)

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*)
(*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,4]]*velocidad2000+demandasADSL1[[i,5]]*velocidad1200+dema
ndasADSL1[[i,6]]*velocidad900+demandasADSL1[[i,7]]*velocidad600+demandasADSL1[[i,8]]*veloci
dad400)*porcentajeBWGarant+demandasADSL1[[i,9]]*velocidadPlus*porcentajeBWGarantPlus},{i,Lo
calidades}]*factrednd;
Print[BWGarant];
BWGarantNuc=Table[({demandasADSL1Nuc[[in,1]]*velocidad2000+demandasADSL1Nuc[[in,2]]*velocid
ad1200+demandasADSL1Nuc[[in,3]]*velocidad900+demandasADSL1Nuc[[in,4]]*velocidad600+demandas
ADSL1Nuc[[in,5]]*velocidad400)*porcentajeBWGarant+demandasADSL1Nuc[[in,6]]*velocidadPlus*po
rcentajeBWGarantPlus},{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["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"];

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["Parámetros de Máximo número de usuarios por componente: Cluster, Shelf y Tarj"];

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["Número de componentes que forman los DSLAMs por ubicación"];

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

Print["numeroClusterFullNuc"];
numeroClusterFullNuc=Floor[sumademandasADSLNuc/maxUsrCluster];
Print[numeroClusterFullNuc];
(* Número de los Clusters completos Nuc *)

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

Print["numeroShelfFull"];
numeroShelfFull=Table[Floor[(sumademandasADSL[[i]]-
numeroClusterFull[[i]]*maxUsrCluster)/maxUsrShelf],{i,Localidades}];

```



```

Print[numeroShelfFull];
(* Número de los Armarios completos *)

Print["numeroShelfFullNuc"];
numeroShelfFullNuc=Floor[(sumademandasADSLNuc-
numeroClusterFullNuc*maxUsrCluster)/maxUsrShelf];
Print[numeroShelfFullNuc];
(* Número de los Armarios completos Nuc *)

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

Print["nuermoTarjLT24"];
numeroTarjLT24=Table[Ceiling[((sumademandasADSL[[i]]-
(numeroClusterFull[[i]]*maxUsrCluster)-
(numeroShelfFull[[i]]*maxUsrShelf))/maxUsrTarj]],{i,Localidades}];
Print[numeroTarjLT24];
(* Número de Tarjetas de Terminación de Línea *)

Print["nuermoTarjLT24Nuc"];
numeroTarjLT24Nuc=Ceiling[(sumademandasADSLNuc-numeroClusterFullNuc*maxUsrCluster-
numeroShelfFullNuc*maxUsrShelf)/maxUsrTarj];
Print[numeroTarjLT24Nuc];
(* Número de Tarjetas de Terminación de Línea Nuc*)

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

Print["numeroArmario"];
numeroArmario=Table[Ceiling[(sumademandasADSL[[i]]-
numeroClusterFull[[i]]*maxUsrCluster)/(maxUsrShelf*2)],{i,Localidades}];
Print[numeroArmario];

Print["numeroArmarioNuc"];
numeroArmarioNuc=Ceiling[(sumademandasADSLNuc-
numeroClusterFull*maxUsrCluster)/(maxUsrShelf*2)];
Print[numeroArmarioNuc];

Print["SumnumeroArmario"];
SumnumeroArmario=Plus@@numeroArmario;
Print[SumnumeroArmario];
(*Instruccion añadida 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[nu
meroShelfFull[[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 añadida 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]]*maxUsrClu
ster+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 añadida 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 añadida 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[(BWClusterFull[[i]]/numeroClusterFull[[i]]]≤
34,34,If[(BWClusterFull[[i]]/numeroClusterFull[[i]]]≤155,155,"No"]]],0)),{i,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 interfases 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 interfases E-3 hacia los PoPs ATM*)

Print["InterfazE1HaciaPOP"];
InterfazE1HaciaPOP=Table[(If[TarjNTClusterParcial[[i]]==
8,Ceiling[BWclusterParcial[[i]]/2.048],0)],{i,Localidades}];
Print[InterfazE1HaciaPOP];

Print["InterfazE1HaciaPOPNUC"];
InterfazE1HaciaPOPNUC=Table[BWclusterFullNuc[[i]]/2.048,{i,nx}];
Print[InterfazE1HaciaPOPNUC];

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

Print["NumE1Equivs"];
NumE1Equivs=Table[63*InterfazSTM1HaciaPOP[[i]]+16*InterfazE3HaciaPOP[[i]]+InterfazE1HaciaPO
P[[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["WPorLineaADSLAM"];
WPorLineaADSLAM=energiaADSL[[2]][[2]];
Print[WPorLineaADSLAM];

(*Calcula el costo del consumo de energia del DSLAM por watio por site*)
Print["CostoEnergiaDSLAM"];
CostoEnergiaDSLAM=Table[CostePorWDSLAM*WPorLineaADSLAM*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["SumCostoEnergiaDSLAM"];
SumCostoEnergiaDSLAM=Plus@@CostoEnergiaDSLAM;
Print[SumCostoEnergiaDSLAM];

Print["REPARTO A SERVICIOS POR DEPARTAMENTO"];
(*Carga de los ficheros de entrada*)
OpenRead["PuertosATM155ParaGigADSL_LIMA6PE3TOT.txt"];
PuertosLima=ReadList["PuertosATM155ParaGigADSL_LIMA6PE3TOT.txt",{Word,Real,
Real,Real,Real,Real}];
Close["PuertosATM155ParaGigADSL_LIMA6PE3TOT.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["CARGA DE LOS FICHEROS DE ENTRADA"];

Print["PorcentajeLinGigADSL"];
PorcentajeLinGigADSL=LineasGigADSL[[1,2]]/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[" Acumula las Demandas a nivel de Departamento y por año "];
(*Se modifiko 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*)

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,SumDemADSL*PorcentajeLinGigADSL/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*)

Print["PorcentLimaGigADSL"];
PorcentLimaGigADSL=If[SumLinLimaGigADSL==0,
  PorcentajesNulos,
  LineasGigADSL[[1,2]]/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["PorcentajealneasGigADSLlLima"];
PorcentajealneasGigADSLlLima=Table[If[CabsLIMA[[j,1]]≠
0, PorcentLimaGigADSL, 0], {j, 1, NumLIMA}];
Print[TableForm[PorcentajealneasGigADSLlLima]];
Print["CÁLCULO DE LOS COSTES de NODOS ATM"];

(*Calcula el numero de usuarios por site para la interfase STM-1*)
Print["UsersSTM1"];
UsersSTM1=Table[Min[(maxUsrCluster*InterfazSTM1HaciaPOP[[i]]), sumademandasADSL[[i]]], {i, Loc
alidades}];
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[SumUserE3];

Print["UserE3"];
UserE3=Table[Prepend[{UserE3[[i]]},xCx[[i]]],{i,Localidades}];
Print[TableForm[UserE3]];

(*Calcula el numero de usuarios por site para la interfase E1*)
Print["UserE1"];
UserE1=Table[sumademandasADSL[[i]]-UsersSTM1[[i,2]]-UserE3[[i,2]],{i,Localidades}];
Print[TableForm[UserE1]];

Print["SumUserE1"];
SumUserE1=Table[Plus@@UserE1];

```

```

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 interfases 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[(UsersE3[[i,1]]≠UsersE3[[i-1,1]]),If[UsersE3[[j,1]]==
UsersE3[[i,1]],InterfazE3[[i]]+=InterfazE3HaciaPOP[[j]],0},{i,2,Localidades},{j,Localidade
s}]
Print[InterfazE3];

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

Print["InterfazE32"];
InterfazE32=0;
Do[If[UsersE3[[j,1]]==
UsersE3[[1,1]],InterfazE32+=InterfazE3HaciaPOP[[j]],0},{j,Localidades}];
Print[InterfazE32];

(*Calcula el numero de interfases 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[(UsersE1[[i,1]]#UsersE1[[i-1,1]]),If[UsersE1[[j,1]]==
UsersE1[[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[UsersE1[[j,1]]==UsersE1[[1,1]],Intfaz2E1+=InterfazE1HaciaPOP[[j]],0},{j,Localidades}]
Print[Intfaz2E1];

(*Calcula el numero de interfases 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[Intfaz3E1[[i]]/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[(Intfaz3E1[[i]]-
ClusterPreconc[[i]]*maxUsrATMCluster)/maxUsrATMShelf},{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[(Intfaz3E1[[i]]-ClusterPreconc[[i]]*maxUsrATMCluster-
ShelfPreconc[[i]]*maxUsrATMShelf)/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[(Intfaz3E1[[i]]-
ClusterPreconc[[i]]*maxUsrATMCluster)/(maxUsrATMShelf*2)],{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]]*precioAr
mario+(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 añadido 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[TarjP
reconc[[i]]≤4,34,155],0]],{i,Localidades}];
Print[TarjNTPreconcClusterParcial];

(*Se ha añadido 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 añadido 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 añadido 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 añadido 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 añadido 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 interfases E3 de 34Mbps en el Nodo ATM por cabecera*)
Print["InterfazE3HaciaATM"];
InterfazE3HaciaATM=Table[TotIntfzPreconcE3[[i]]+InterfazE33[[i]],{i,Localidades}];
Print[InterfazE3HaciaATM];

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

(*Precios para Nodos ATM en la etapa de concentración*)
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 concentración*)
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[DemPorPOP3[[i]]/16000],{i,Localidades}];
Print[NumNodosATM];

(*Se ha añadido 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[InterfazSTM1HaciaATM[[i]]/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[InterfazE3HaciaATM[[i]]/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+NumTar
jE3Conc[[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]],{l,NumDP
TOS-NumLIMA}]

```

```

Print[TableForm[PorcentajealneasGigADSL Lima2]];

Print["PorclneasGigADSL Lima22"];
PorclneasGigADSL Lima22=Length[PorcentajealneasGigADSL Lima2];
Print[PorclneasGigADSL Lima22];
PorcentajealneasGigADSL Lima2;

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]],{1,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]],
            PorcentajealneasGigADSL Lima2[[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]];

(*Calcula el costo total del Punto de Presencia (PoP) ATM por cabecera*)
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["PARAMETROS COSTES"];
```

```
(*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=Wacc/(1-TasaImpositiva);
```

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

```
FactAnualizCxDSLAM=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortCxDSLAM)));
FactAnualizCxNodoATM=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortCxNodoATM)));
FactAnualizTxEquipos=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortTxEquipos)));
FactAnualizTxPtaExt=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortTxPtaExt)));
FactAnualizTxPortador=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortTxPortador)));
FactAnualizPtaSecENERGIA=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortPtaSecENERGIA)));
FactAnualizPtaSecINMOB=CosteCapital/(1-((1+CosteCapital)^(-AñosAmortPtaSecINMOB)));
```

```
Print["EVOLUCION PRECIOS"];
```

```
(*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,1,6,2,6,3,6,4,6,5,6,6,6,7,6,8,6,9,14,10,6,11,14,12,14,13,14,14,14,15,14,16,14,17,14,18,19,19,19,20,19,21,6,22,19,23,19,24,19];
```

```
    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]]}]];
    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["TOTALES1ADSL"];
TOTALES1ADSL = 0;
Do[TOTALES1ADSL = TOTALES1ADSL + CADSL[[iads1]]; {iads1, nadsl}];
Print[TOTALES1ADSL];

e1remotehostadslx = zerox;
e1hosttandemadslx = zerox;

Print["e1remotehostadslx"];
Do[If[remote1x[[iads1]] == 1,
  e1remotehostadslx[[iads1]] = e1remotehostadslx[[iads1]] + CADSL[[iads1]];
  {iads1, nadsl}];
Print[e1remotehostadslx]

Print["e1hosttandemadslx"];
Do[If[remote1x[[iads1]] == 1,
  e1hosttandemadslx[[allhostx[[iads1]]]] =
  e1hosttandemadslx[[allhostx[[iads1]]]] + CADSL[[iads1]];
  If[remote1x[[iads1]] == 0 && tandem1x[[iads1]] == 0,
    e1hosttandemadslx[[iads1]] = e1hosttandemadslx[[iads1]] + CADSL[[iads1]];
    {iads1, nadsl}];

Do[If[limax[[iads1]] && esAnillodx[[iads1]] == 2 && hostx[[iads1]],
  If[CxATMLimax[[iads1]] != 1,
    e1hosttandemadslx[[iads1]] = Ceiling[e1hosttandemadslx[[iads1]] * 1.75];
    e1hosttandemadslx[[iads1]] = Ceiling[e1hosttandemadslx[[iads1]] * 0.75];
    {iads1, nadsl}];
  (* Esta instrucción sirve para utilizar el 75% más de Tx para ADSL en anillo de Lima *)
  Print[e1hosttandemadslx];

```

```

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

```

```
factcorr = 1;
```

```
PRINT["CIRCUITOS RANGO A"]
```

```
PRINT["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES"]
```

```
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["CALQUILADOA1"]
CALQUILADOA1= N[ReadList["input_C_AlquiladosA1.txt",Number,RecordLists->True]];
Print[CALQUILADOA1]
(* 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] ] *)

Print["nalqA1"]
nalqA1=Length[CALQUILADOA1];
Print[nalqA1]
(* Identifica el número de elementos de la lista CALQUILADOA1 *)

e1remotehostalqxA1=zerox;
e1hosttandemalqxA1=zerox;
TOTALE1ALQA1=0;

Print["TOTALE1ALQA1"]
Do[TOTALE1ALQA1=TOTALE1ALQA1+CALQUILADOA1[[ialq,2]]/2;
,{ialq,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["CIRCUITOS INTRADEPARTAMENTALES"]

```
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 E1s *)
```

```
Print["CALQUILADOA12"]
CALQUILADOA12= 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[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]];
(* 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 *)

Do[e1remotehostalqxA1[[iax]]=Ceiling[e1remotehostalqxA1[[iax]]*(factcorr)];
e1hosttandemalqxA1[[iax]]=Ceiling[e1hosttandemalqxA1[[iax]]*(factcorr)];,{iax,nx}}

PRINT["CIRCUITOS RANGO B"]

PRINT["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES"]

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= 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=zeron;
e1hosttandemalqxB1=zeron;
TOTALE1ALQB1=0;

Print["TOTALE1ALQB1"]
Do[TOTALE1ALQB1= TOTALE1ALQB1+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 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["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["CIRCUITOS INTRADEPARTAMENTALES"]

```
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=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[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 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 e1remotehostalqxB1 *)

```
Print["e1hosttandemalqxB1"]
Do[aux1=CALQUILADOB12[[ialq,1]];aux2=CALQUILADOB12[[ialq,2]];
    (* 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 *)
```

```
Do[e1remotehostalqxB1[[iax]]=Ceiling[e1remotehostalqxB1[[iax]]*(factcorr)];
e1hosttandemalqxB1[[iax]]=Ceiling[e1hosttandemalqxB1[[iax]]*(factcorr)];,{iax,nx}]
```

```
PRINT["CIRCUITOS RANGO C "]
```

```
PRINT["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES"]
```

```
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=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=zeror;
```

```
e1hosttandemalqxC1=zeror;
```

```
TOTALE1ALQC1=0;
```

```
Print["TOTALE1ALQC1"]
```

```
Do[TOTALE1ALQC1=TOTALE1ALQC1+CALQUILADOC1[[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 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["e1hosttandemalqxC1"]
```

```
Do[If[remote1x[[CALQUILADOC1[[ialq,1]]]]==1,
e1hosttandemalqxC1[[allhostx[[CALQUILADOC1[[ialq,1]]]]]]+=
e1hosttandemalqxC1[[allhostx[[CALQUILADOC1[[ialq,1]]]]]]+CALQUILADOC1[[ialq,2]]];
If[remote1x[[CALQUILADOC1[[ialq,1]]]]==0&&tandem1x[[CALQUILADOC1[[ialq,1]]]]==0,
e1hosttandemalqxC1[[CALQUILADOC1[[ialq,1]]]]+=
e1hosttandemalqxC1[[CALQUILADOC1[[ialq,1]]]]+CALQUILADOC1[[ialq,2]]];,{ialq,nalqC1}];
```

```
Print[e1hosttandemalqxC1]
```

```
(* 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["CIRCUITOS INTRADEPARTAMENTALES"]
```

```
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=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["e1remotehostalqxC1"]
Do[aux1=CALQUILADOC12[[ialq,1]];
  aux2=CALQUILADOC12[[ialq,2]];
  If[remote1x[[aux1]]==1,e1remotehostalqxC1[[aux1]]=e1remotehostalqxC1[[aux1]]+CALQUILADOC12[[ialq,3]]];;
  If[remote1x[[aux2]]==1,e1remotehostalqxC1[[aux2]]=e1remotehostalqxC1[[aux2]]+CALQUILADOC12[[ialq,3]]];;
  ,{ialq,nalqC12}];
Print[e1remotehostalqxC1]
(* 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 e1remotehostalqxC1 *)

Print["e1hosttandemalqxC1"]
Do[aux1=CALQUILADOC12[[ialq,1]];aux2=CALQUILADOC12[[ialq,2]];
(* 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,e1hosttandemalqxC1[[aux2]]=e1hosttandemalqxC1[[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,e1hosttandemalqxC1[[aux1]]=e1hosttandemalqxC1[[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 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[e1hosttandemalqxC1]
(* Este desarrollo esta identificando los E1s 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 *)

Do[e1remotehostalqxC1[[iax]]=Ceiling[e1remotehostalqxC1[[iax]]*(factcorr)];
e1hosttandemalqxC1[[iax]]=Ceiling[e1hosttandemalqxC1[[iax]]*(factcorr)];,{iax,nx}}

```

PRINT["CIRCUITOS RANGO A, B y C OTROS"]

PRINT["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES"]

```

OpenRead["input_C_AlquiladosABC2.txt"];
(* Leyendo la información de los otros circuitos de rango A, B y C 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["CALQUILADOABC2"]
CALQUILADOABC2= N[ReadList["input_C_AlquiladosABC2.txt",Number,RecordLists→True]];
Print[CALQUILADOABC2]
(* Presenta la información del archivo de entrada de otros circuitos de rango A, B y 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["nalqABC2"]
nalqABC2=Length[CALQUILADOABC2];
Print[nalqABC2]
(* Identifica el número de elementos de la lista CALQUILADOABC2 *)

```

```

e1remotehostalqxABC2=zerox;
e1hosttandemalqxABC2=zerox;
TOTALE1ALQABC2=0;

Print["TOTALE1ALQABC2"]
Do[TOTALE1ALQABC2=TOTALE1ALQABC2+CALQUILADOABC2[[ialq,2]]/2;,{ialq,nalqABC2}];
Print[TOTALE1ALQABC2]
(* Identifica el total de E1s asociados a otros circuitos Interdepartamentales de rango A, B y 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["e1remotehostalqxABC2"]
Do[If[remote1x[[CALQUILADOABC2[[ialq,1]]]]==1,
    e1remotehostalqxABC2[[CALQUILADOABC2[[ialq,1]]]]+e1remotehostalqxABC2[[CALQUILADOABC2[[ialq,1]]]]+
    CALQUILADOABC2[[ialq,2]]];,{ialq,nalqABC2}];
Print[e1remotehostalqxABC2]
(* Esta identificando los E1s correspondientes a otros circuitos interdepartamentales del tipos A, B y 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["e1hosttandemalqxABC2"]
Do[If[remote1x[[CALQUILADOABC2[[ialq,1]]]]==1,
    e1hosttandemalqxABC2[[allhostx[[CALQUILADOABC2[[ialq,1]]]]]]+=
    e1hosttandemalqxABC2[[allhostx[[CALQUILADOABC2[[ialq,1]]]]]]+CALQUILADOABC2[[ialq,2]]];;
    If[remote1x[[CALQUILADOABC2[[ialq,1]]]]==0&&andem1x[[CALQUILADOABC2[[ialq,1]]]]==0,
        e1hosttandemalqxABC2[[CALQUILADOABC2[[ialq,1]]]]+=
        e1hosttandemalqxABC2[[CALQUILADOABC2[[ialq,1]]]]+CALQUILADOABC2[[ialq,2]]];;
    ,{ialq,nalqABC2}];
Print[e1hosttandemalqxABC2]
(* Esta identificando los E1s correspondientes a otros circuitos interdepartamentales del tipo A, B y 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 y que no son
Tandem y para cada una de ellas se realiza la sumatoria del total de E1s (2do elemento de cada sublista) *)

PRINT["CIRCUITOS INTRADEPARTAMENTALES"]

OpenRead["input_C_Alquilados2ABC2.txt"];
(* Leyendo la información de otros circuitos alquilados del rango A, B y C 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 E1s *)

Print["CALQUILADOABC22"]
CALQUILADOABC22= N[ReadList["input_C_Alquilados2ABC2.txt",Number,RecordLists->True]];
Print[CALQUILADOABC22]
(* Presenta la información del archivo de entrada de otros circuitos alquilados del rango A, B y 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["nalqABC22"]
nalqABC22=Length[CALQUILADOABC22];
Print[nalqABC22]
(* Identifica el número de elementos de la lista CALQUILADOABC22 *)

Print["TOTALE1ALQABC2"]
Do[TOTALE1ALQABC2=TOTALE1ALQABC2+CALQUILADOABC22[[ialq,3]];,{ialq,nalqABC22}];
Print[TOTALE1ALQABC2]
(* Identifica el total de E1s asociados a otros circuitos Interdepartamentales del tipo A, B y 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 TOTALE1ALQA2 que correspondió a los
E1s asociados a otros circuitos Interdepartamentales del tipo A, B y C que eran atribuibles como cargas en
el tramo intradepartamental *)

Print["e1remotehostalqxABC2"]
Do[aux1=CALQUILADOABC22[[ialq,1]];
    aux2=CALQUILADOABC22[[ialq,2]];
    If[remote1x[[aux1]]==1,
        e1remotehostalqxABC2[[aux1]]=e1remotehostalqxABC2[[aux1]]+CALQUILADOABC22[[ialq,3]]];;
    If[remote1x[[aux2]]==1,
        e1remotehostalqxABC2[[aux2]]=e1remotehostalqxABC2[[aux2]]+CALQUILADOABC22[[ialq,3]]];;
    ,{ialq,nalqABC22}];
Print[e1remotehostalqxABC2]
(* Esta identificando los E1s correspondientes a otros circuitos Intradepartamentales del tipo A, B y C 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 e1remotehostalqxABC2 *)

Print["e1hosttandemalqxABC2"]
Do[aux1=CALQUILADOABC22[[ialq,1]];
    aux2=CALQUILADOABC22[[ialq,2]];

```



```
(* 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,
  e1hosttandemalqxABC2[[aux2]]=e1hosttandemalqxABC2[[aux2]]+CALQUILADOABC22[[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,
  e1hosttandemalqxABC2[[aux1]]=e1hosttandemalqxABC2[[aux1]]+CALQUILADOABC22[[ialq,3]]];;
,{ialq,nalqABC22}];
(* 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[e1hosttandemalqxABC2]
(* Este desarrollo esta identificando los E1s correspondientes a circuitos Intradepartamentales distintos
del tipo A y 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 e1hosttandemalqxABC2 *)
```

```
Do[e1remotehostalqxABC2[[iax]]=Ceiling[e1remotehostalqxABC2[[iax]]*(factcorr)];
  e1hosttandemalqxABC2[[iax]]=Ceiling[e1hosttandemalqxABC2[[iax]]*(factcorr)];,{iax,nx}]
```

PRINT["CIRCUITOS NO OPERADORES"]

```
PRINT["TRAMOS INTRADEPARTAMENTALES DE LOS CIRCUITOS INTERDEPARTAMENTALES"]
```

```
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= 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 E1s]] *)
```

```
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+CALQUILADOABCnop[[ialq,2]]/2;,{ialq,nalqABCnop}];
```

```
Print[TOTALE1ALQABCnop]
```

```
(* Identifica el total de E1s 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 E1s 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 E1s 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 E1s (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 E1s 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 E1s (2do elemento de cada sublista) *)

```
PRINT["CIRCUITOS INTRADEPARTAMENTALES"]
```

```
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 E1s *)
```

```
Print["CALQUILADOABC2nop"]
```

```
CALQUILADOABC2nop=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 E1s] ] *)
```

```
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 E1s asociados a los circuitos alquilados Interdepartamentales de los no operadores. 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 TOTALE1ALQB2 que correspondió a los E1s 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[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 origen o destino a cada central remota. Si se optó por estimar los costos de los circuitos 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]];
```

```
(* 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 *)
```

```
Do[e1remotehostalqxABCnop[[iax]]=Ceiling[e1remotehostalqxABCnop[[iax]]*(factcorr)];
```

```
e1hosttandemalqxABCnop[[iax]]=Ceiling[e1hosttandemalqxABCnop[[iax]]*(factcorr)];,{iax,nx}]
```

```
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=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["ALGORITMO DE PRIM"]
```

```
(***** en estrella con fibra enterrada *****)
```

```

If[ConsideraPRIM≠0,
  nodo=zer0;
  numeronodos=0;
  prime1remotex=zer0;      (* a nivel de voz *)
  primloadremotexLocx=zer0; (* Voz Local *)
  primloadremotexLDNx=zer0; (* Voz LDN *)
  primloadremotexLDIx=zer0; (* Voz LDI *)
  prime1remotexA1=zer0;     (* circuitos tipo A *)
  prime1remotexB1=zer0;     (* circuitos tipo B *)
  prime1remotexC1=zer0;     (* circuitos tipo C *)
  prime1remotexABC2=zer0;   (* otros circuitos tipo A, B y C *)
  prime1remotexABCnop=zer0; (* circuitos de No Operadores *)
  prime1remotexitxx=zer0;   (* circuitos de Interconexión *)
  prime1remotexadslx=zer0;  (* Los de ADSL *)
  Do[If[host1x[[iax]]==1,
    numeronodos=1;
    nodo[[numeronodos]]=iax,
    If[(esAnillox[[iax]]==0)&&((transtechx[[iax]]==Enterrado)|| (transtechx[[iax]]==Radio)),
      numeronodos++;
      nodo[[numeronodos]]=iax;]],];
  If[(host1x[[iax+1]]==1)&&(numeronodos>2), (* comienzo Prim *)
    Print["Prim"];
    disttotal =0;
    nodoconectado=Table[0,{ix,numeronodos}];
    nodoconectado[[1]]=1; (* vale 1 si ya forma parte del grafo *)
    aquienmeconecto=zer0;
    nodoencontrado=1;
    While[nodoencontrado≠0,
      distaux=0;
      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)&&
              ((transtechx[[nodo[[iinod]]]]==Enterrado)|| (host1x[[nodo[[iinod]]]]==1))||
              (transtechx[[nodo[[jjnod]]]]==Radio)))]],
            ((nodoconectado[[jjnod]]==1)&&(nodoconectado[[iinod]]==0)&&
              ((transtechx[[nodo[[jjnod]]]]==Enterrado)|| (transtechx[[nodo[[iinod]]]]==
Radio))))),
          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;
        disthostx[[nodo[[nodoencontrado]]]]=distaux/factorNoLinealNoUrban;
        repfibrax[[nodo[[nodoencontrado]]]]=distaux/factorNoLinealNoUrban;
        Conexion[[nodo[[nodoencontrado]]]=nodo[[nodoalqueseconecta]];
        If[(nodoalqueseconecta==1)&&(ConsideraPRIM==2)&&(transtechx[[nodo[[nodoencontrado]]]]==
Enterrado),
          disthostx[[nodo[[nodoencontrado]]]]=2*disthostx[[nodo[[nodoencontrado]]]];
          repfibrax[[nodo[[nodoencontrado]]]]=2*repfibrax[[nodo[[nodoencontrado]]]];];
        If[(ConsideraPRIM==3)&&(transtechx[[nodo[[nodoencontrado]]]]==Enterrado),
          disthostx[[nodo[[nodoencontrado]]]]=2*disthostx[[nodo[[nodoencontrado]]]];
          repfibrax[[nodo[[nodoencontrado]]]]=2*repfibrax[[nodo[[nodoencontrado]]]];];
        aquienmeconecto[[nodoencontrado]]=nodoalqueseconecta;];
        disttotal=disttotal+distaux/factorNoLinealNoUrban;];
    (* ahora actualizar numero de e1s *)
    Do[prime1remotex[[ix]]=e1remotex[[ix]];
      primloadremotexLocx[[ix]]=loadremotexLocx[[ix]];
      primloadremotexLDNx[[ix]]=loadremotexLDNx[[ix]];
      primloadremotexLDIx[[ix]]=loadremotexLDIx[[ix]];
      prime1remotexA1[[ix]]=e1remotexA1[[ix]];
      prime1remotexB1[[ix]]=e1remotexB1[[ix]];
      prime1remotexC1[[ix]]=e1remotexC1[[ix]];
      prime1remotexABC2[[ix]]=e1remotexABC2[[ix]];
      prime1remotexABCnop[[ix]]=e1remotexABCnop[[ix]];
      prime1remotexitxx[[ix]]=e1remotexitxx[[ix]];
      prime1remotexadslx[[ix]]=e1remotexadslx[[ix]],{ix,nx}];
    aquienmeconecto2=aquienmeconecto;
    For[iinod=2,iinod<numeronodos+1,
      While[(aquienmeconecto2[[iinod]]!=0) && (aquienmeconecto2[[iinod]]!=1),
        e1remotex[[nodo[aquienmeconecto2[[iinod]]]]]=
        e1remotex[[nodo[aquienmeconecto2[[iinod]]]]]+prime1remotex[[nodo[[iinod]]]];
        loadremotexLocx[[nodo[aquienmeconecto2[[iinod]]]]]=
        loadremotexLocx[[nodo[aquienmeconecto2[[iinod]]]]]+primloadremotexLocx[[nodo[[iinod]]]];
        loadremotexLDNx[[nodo[aquienmeconecto2[[iinod]]]]]=
        loadremotexLDNx[[nodo[aquienmeconecto2[[iinod]]]]]+primloadremotexLDNx[[nodo[[iinod]]]];
        loadremotexLDIx[[nodo[aquienmeconecto2[[iinod]]]]]=
        loadremotexLDIx[[nodo[aquienmeconecto2[[iinod]]]]]+primloadremotexLDIx[[nodo[[iinod]]]];];
  ]

```



```

e1remotehostalqxAl[[nodo[[a quien me conecto 2[[iinod]]]]]]]=
e1remotehostalqxAl[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostalqxAl[[nodo[[iinod]]]];
e1remotehostalqxB1[[nodo[[a quien me conecto 2[[iinod]]]]]]]=
e1remotehostalqxB1[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostalqxB1[[nodo[[iinod]]]];
e1remotehostalqxC1[[nodo[[a quien me conecto 2[[iinod]]]]]]]=
e1remotehostalqxC1[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostalqxC1[[nodo[[iinod]]]];
e1remotehostalqxABC2[[nodo[[a quien me conecto 2[[iinod]]]]]]]=

e1remotehostalqxABC2[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostalqxABC2[[nodo[[iinod]]]];
e1remotehostalqxABCnop[[nodo[[a quien me conecto 2[[iinod]]]]]]]=

e1remotehostalqxABCnop[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostalqxABCnop[[nodo[[iinod]]]];
e1remotehostadslx[[nodo[[a quien me conecto 2[[iinod]]]]]]]=
e1remotehostadslx[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostadslx[[nodo[[iinod]]]];
e1remotehostitxx[[nodo[[a quien me conecto 2[[iinod]]]]]]]=
e1remotehostitxx[[nodo[[a quien me conecto 2[[iinod]]]]]]+prime1remotehostitxx[[nodo[[iinod]]]];

a quien me conecto 2[[iinod]]=a quien me conecto[[a quien me conecto 2[[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[MatrixForm[{e1remotehostx,loadremotehostLocx,loadremotehostLDNx,loadremotehostLDIx,loadremotehostx,e1re
motehostalqxAl,e1remotehostalqxB1,e1remotehostalqxC1,

e1remotehostalqxABC2,e1remotehostalqxABCnop,e1remotehostadslx,e1remotehostitxx},TableDirections-
>{Row,Column},TableHeadings->{
{"E1s Voz","Erlangs Local","Erlangs LDN","Erlangs LDI","Erlangs Total","E1s Rango A","E1s
Rango B","E1s Rango C","E1s Otros","E1s No Op",
"E1s ADSL","E1s ITX"},None}]];

```

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 *)
e1AnilloalqpABC2=zerop; (* Otros Circuitos Rango A, B y C *)
e1AnilloalqpABCnop=zerop; (* Circuitos de No Operadores *)
e1AnilloalqpC1=zerop; (* Circuitos Rango C *)
e1Anilloadslp=zerop; (* ADSL *)
e1Anilloitxp=zerop; (* Circuitos de Interconexión *)
ProvAnillo=1;
esAnillo=0;

Print["e1Anillop"]
Do[If[host1x[[iax]]==1,
  If[esAnillop[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
  If[esAnillo==1,
    e1Anillop[[ProvAnillo]]=e1Anillop[[ProvAnillo]]+If[esAnillop[[iax]]==0,0,1]*e1remotehostx[[iax]]];
  ,{iax,nx}];
Print[e1Anillop]
(* Se estima el número de E1s a nivel de cada anillo provincia para voz. 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 *)

Print["loadAnilloLocp"]
Do[If[host1x[[iax]]==1,
  If[esAnillop[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
  If[esAnillo==1,
    loadAnilloLocp[[ProvAnillo]]=loadAnilloLocp[[ProvAnillo]]+If[esAnillop[[iax]]==
0,0,1]*loadremotehostLocx[[iax]]];
  ,{iax,nx}];
Print[loadAnilloLocp]
(* Se estima la carga en Erlangs a nivel de cada anillo provincial para voz Local. 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
loadremotehostLocx *)

Print["loadAnilloLDNp"]
Do[If[host1x[[iax]]==1,
  If[esAnillop[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
  If[esAnillo==1,
    loadAnilloLDNp[[ProvAnillo]]=loadAnilloLDNp[[ProvAnillo]]+If[esAnillop[[iax]]==
0,0,1]*loadremotehostLDNx[[iax]]];
  ,{iax,nx}];
Print[loadAnilloLDNp]
(* Se estima la carga en Erlangs a nivel de cada anillo provincial para voz LDN. 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 loadremotehostLDNx *)

Print["loadAnilloLDIp"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        loadAnilloLDIp[[ProvAnillo]]=loadAnilloLDIp[[ProvAnillo]]+If[esAnillopx[[iax]]==
0,0,1]*loadremotehostLDIx[[iax]]];];
    ,{iax,nx}];
```

Print[loadAnilloLDIp]

(* Se estima la carga en Erlangs a nivel de cada anillo provincial para voz LDI. 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 loadremotehostLDIx *)

Print["loadAnilloLp"]

loadAnilloLp=loadAnilloLocp+loadAnilloLDNp+loadAnilloLDIp;

Print[loadAnilloLp]

(* Se estima la carga en Erlangs a nivel de cada anillo provincial para voz Total. 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 loadremotehostx *)

Print["e1AnilloalqpA1"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1AnilloalqpA1[[ProvAnillo]]=e1AnilloalqpA1[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostalqxA1[[iax]]];];
    ,{iax,nx}];
```

Print[e1AnilloalqpA1]

(* Se estima el número de E1s a nivel de cada anillo provincia para circuitos de Rango A. 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 e1remotehostalqxA1 *)

Print["e1AnilloalqpB1"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1AnilloalqpB1[[ProvAnillo]]=e1AnilloalqpB1[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostalqxB1[[iax]]];];
    ,{iax,nx}];
```

Print[e1AnilloalqpB1]

(* Se estima el número de E1s a nivel de cada anillo provincia para circuitos de Rango B. 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 e1remotehostalqxB1 *)

Print["e1AnilloalqpC1"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1AnilloalqpC1[[ProvAnillo]]=e1AnilloalqpC1[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostalqxC1[[iax]]];];
    ,{iax,nx}];
```

Print[e1AnilloalqpC1]

(* Se estima el número de E1s a nivel de cada anillo provincia para circuitos de Rango C. 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 e1remotehostalqxC1 *)

Print["e1AnilloalqpABC2"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1AnilloalqpABC2[[ProvAnillo]]=e1AnilloalqpABC2[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostalqxABC2[[iax]]];];
    ,{iax,nx}];
```

Print[e1AnilloalqpABC2]

(* Se estima el número de E1s a nivel de cada anillo provincia para otros circuitos de Rango A, B y C. 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 e1remotehostalqxABC2 *)

Print["e1AnilloalqpABCnop"]

```
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1AnilloalqpABCnop[[ProvAnillo]]=e1AnilloalqpABCnop[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostalqxABCnop[[iax]]];];
    ,{iax,nx}];
```

Print[e1AnilloalqpABCnop]

(* Se estima el número de E1s a nivel de cada anillo provincia para circuitos de Rango A, B y C No Operadores. 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 e1remotehostalqxABCnop *)

Print["e1AnilloloadsIp"]

```

Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1Anilloads1p[[ProvAnillo]]=e1Anilloads1p[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostads1x[[iax]]];;, {iax,nx}];
Print[e1Anilloads1p]
(* Se estima el número de E1s a nivel de cada anillo provincia para ADSL. 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
e1remotehostads1x *)

Print["e1Anilloitxp"]
Do[If[host1x[[iax]]==1,
    If[esAnillopx[[iax]]==1,ProvAnillo=provx[[iax]];esAnillo=1,esAnillo=0]];
    If[esAnillo==1,
        e1Anilloitxp[[ProvAnillo]]=e1Anilloitxp[[ProvAnillo]]+
        If[esAnillopx[[iax]]==0,0,1]*e1remotehostitxx[[iax]]];;, {iax,nx}];
Print[e1Anilloitxp]
(* Se estima el número de E1s a nivel de cada anillo provincia para el tipo de servicio que se haya elegido.
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 e1remotehostitxx *)

PRINT["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 *)
e1AnilloalqdABC2=zerod;  (* Otros Circuitos Rango A, B y C *)
e1AnilloalqdABCnop=zerod;(* Circuitos de No Operadores *)
e1Anilloads1d=zerod;     (* ADSL *)
e1AnilloRDd=zerod;      (* Rutas directas *)
e1Anilloitxd=zerod;     (* Circuitos de Interconexión *)
DepAnillo=0;
esAnillo=0;

Print["e1Anillod"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;esAnillo=0;]];
        If[esAnillo==1,
            e1Anillod[[DepAnillo]]=e1Anillod[[DepAnillo]]+
            If[esAnillodx[[AnilloCabeceras[[iax]]]]==NA,0,1]*(e1hosttandemx[[AnilloCabeceras[[iax]]]]);;];
    , {iax,np}]
Print[e1Anillod]
(* Se estima el número de E1s de cada anillo departamental para voz. 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 *)

Print["loadAnilloLocd"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;esAnillo=0;]];
        If[esAnillo==1,
            loadAnilloLocd[[DepAnillo]]=loadAnilloLocd[[DepAnillo]]+
            If[esAnillodx[[AnilloCabeceras[[iax]]]]==
NA,0,1]*(loadhosttandemLocx[[AnilloCabeceras[[iax]]]]);;];
    , {iax,np}]
Print[loadAnilloLocd]
(* Se estima la carga en Erlangs de cada anillo departamental para voz Local. 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
loadhosttandemLocx *)

Print["loadAnilloLDNd"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;esAnillo=0;]];
        If[esAnillo==1,
            loadAnilloLDNd[[DepAnillo]]=loadAnilloLDNd[[DepAnillo]]+
            If[esAnillodx[[AnilloCabeceras[[iax]]]]==
NA,0,1]*(loadhosttandemLDNx[[AnilloCabeceras[[iax]]]]);;];
    , {iax,np}]
Print[loadAnilloLDNd]
(* Se estima la carga en Erlangs de cada anillo departamental para voz LDN. 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
loadhosttandemLDNx *)

Print["loadAnilloLDId"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,

```

```

        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
    If[esAnillo==1,
        loadAnilloLDId[[DepAnillo]]=loadAnilloLDId[[DepAnillo]]+
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]]=
NA,0,1]*(loadhosttandemLDIx[[AnilloCabeceras[[iax]]]]);];];
    ,{iax,np}}
Print[loadAnilloLDId]
(* Se estima la carga en Erlangs de cada anillo departamental para voz LDI. 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
loadhosttandemLDIx *)

Print["loadAnillod"]
loadAnillod=loadAnilloLocd+loadAnilloLDNd+loadAnilloLDId;
Print[loadAnillod]
(* Se estima la carga en Erlangs de cada anillo departamental para voz Total. 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
loadhosttandemx *)

Print["e1AnilloalqdA1"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
        If [esAnillo==1,
            e1AnilloalqdA1[[DepAnillo]]=e1AnilloalqdA1[[DepAnillo]]+
            If[esAnilloDx[[AnilloCabeceras[[iax]]]]]=
NA,0,1]*(e1hosttandemalqxA1[[AnilloCabeceras[[iax]]]]);];];
    ,{iax,np}}
Print[e1AnilloalqdA1]
(* Se estima el número de E1s de cada anillo departamental para circuitos Rango A. 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 e1hosttandemalqxA1 *)

Print["e1AnilloalqdB1"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
        If [esAnillo==1,
            e1AnilloalqdB1[[DepAnillo]]=e1AnilloalqdB1[[DepAnillo]]+
            If[esAnilloDx[[AnilloCabeceras[[iax]]]]]=
NA,0,1]*(e1hosttandemalqxB1[[AnilloCabeceras[[iax]]]]);];];
    ,{iax,np}}
Print[e1AnilloalqdB1]
(* Se estima el número de E1s de cada anillo departamental para circuitos Rango B. 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 e1hosttandemalqxB1 *)

Print["e1Anilloalqdc1"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
        If [esAnillo==1,
            e1Anilloalqdc1[[DepAnillo]]=e1Anilloalqdc1[[DepAnillo]]+
            If[esAnilloDx[[AnilloCabeceras[[iax]]]]]=
NA,0,1]*(e1hosttandemalqxC1[[AnilloCabeceras[[iax]]]]);];];
    ,{iax,np}}
Print[e1Anilloalqdc1]
(* Se estima el número de E1s de cada anillo departamental para circuitos Rango C. 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 e1hosttandemalqxC1 *)

Print["e1AnilloalqdABC2"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
        If [esAnillo==1,
            e1AnilloalqdABC2[[DepAnillo]]=e1AnilloalqdABC2[[DepAnillo]]+
            If[esAnilloDx[[AnilloCabeceras[[iax]]]]]=
NA,0,1]*(e1hosttandemalqxABC2[[AnilloCabeceras[[iax]]]]);];];
    ,{iax,np}}
Print[e1AnilloalqdABC2]
(* Se estima el número de E1s de cada anillo departamental para otros circuitos Rango A, B y C. 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 e1hosttandemalqxABC2 *)

Print["e1AnilloalqdABCnop"]
Do[If[AnilloCabeceras[[iax]]!=0,
    If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
        If[esAnilloDx[[AnilloCabeceras[[iax]]]]!= NA,
            DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
        If [esAnillo==1,

```

```

e1AnilloalqdABCnop[[DepAnillo]]=e1AnilloalqdABCnop[[DepAnillo]]+
  If[esAnillodx[[AnilloCabeceras[[iax]]]]==
NA,0,1]*(e1hosttandemalqxABCnop[[AnilloCabeceras[[iax]]]]);];];
,{iax,np}}
Print[e1AnilloalqdABCnop]
(* Se estima el número de E1s de cada anillo departamental para circuitos Rango A, B y C de los no
operadores. 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 e1hosttandemalqxABCnop *)

Print["e1Anilloadsld"]
Do[If[AnilloCabeceras[[iax]]!=0,
  If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
    If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
      DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
  If [esAnillo=1,
e1Anilloadsld[[DepAnillo]]=e1Anilloadsld[[DepAnillo]]+
  If[esAnillodx[[AnilloCabeceras[[iax]]]]==NA,0,1]*(e1hosttandemadslx[[AnilloCabeceras[[iax]]]]);];];
,{iax,np}}
Print[e1Anilloadsld]
(* Se estima el número de E1s de cada anillo departamental para ADSL. 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
e1hosttandemadslx *)

Print["e1Anilloitxd"]
Do[If[AnilloCabeceras[[iax]]!=0,
  If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
    If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
      DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
  If [esAnillo=1,
e1Anilloitxd[[DepAnillo]]=e1Anilloitxd[[DepAnillo]]+
  If[esAnillodx[[AnilloCabeceras[[iax]]]]==NA,0,1]*(e1hosttandemitxx[[AnilloCabeceras[[iax]]]]);];];
,{iax,np}}
Print[e1Anilloitxd]
(* Se estima el número de E1s de cada anillo departamental para Circuitos de Interconexión. 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 e1hosttandemitxx *)

Print["e1AnilloRDd"]
Do[If [AnilloCabeceras[[iax]]!=0,
  If[tandem1x[[AnilloCabeceras[[iax]]]]==1,
    If[esAnillodx[[AnilloCabeceras[[iax]]]]!= NA,
      DepAnillo=deptx[[AnilloCabeceras[[iax]]]];esAnillo=1;;esAnillo=0;];];
  If [esAnillo=1,
e1AnilloRDd[[DepAnillo]]=e1AnilloRDd[[DepAnillo]]+If[esAnillodx[[AnilloCabeceras[[iax]]]]==
NA,0,1]*e1RDx[[AnilloCabeceras[[iax]]]]];];];,{iax,np}}
Print[e1AnilloRDd]
(* Se estima el número de E1s de cada anillo departamental para rutas Directas. 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 e1RDx *)

PRINT["A NIVEL DE TODOS LOS SERVICIOS"]

Print["e1remotehostTOTALx"]
e1remotehostTOTALx=Ceiling[e1remotehostx+e1remotehostadslx+e1remotehostalqxA1+e1remotehostalqxABC2+e1remoteho
stalqxB1+e1remotehostalqxABCnop+e1remotehostalqxC1+e1remotehostitxx];
Print[e1remotehostTOTALx]
(* Estima el total de E1s a nivel RH 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["e1hosttandemTOTALx"]
e1hosttandemTOTALx=Ceiling[e1hosttandemx+e1hosttandemadslx+e1hosttandemalqxA1+e1hosttandemalqxABC2+e1hosttand
emalqxB1+e1hosttandemalqxABCnop+e1hosttandemalqxC1+e1hosttandemitxx];
Print[e1hosttandemTOTALx]
(* Estima el total de E1s a nivel 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["e1AnilloTOTALd"]
e1AnilloTOTALd=Ceiling[e1Anillod+e1AnilloalqdA1+e1AnilloalqdABC2+e1AnilloalqdB1+e1AnilloalqdABCnop+e1Anilloal
qdC1+e1Anilloadsld+e1Anilloitxd];
Print[e1AnilloTOTALd]
(* Estima el total de E1s de los anillos departamentales para todos los servicios. Para ellos suma los E1s
asociados a cada uno de los servicios. *)

Print["e1AnilloTOTALp"]
e1AnilloTOTALp=Ceiling[e1Anillop+e1AnilloalqpA1+e1AnilloalqpABC2+e1AnilloalqpB1+e1AnilloalqpABCnop+e1Anilloal
qpC1+e1Anilloadslp+e1Anilloitxp];
Print[e1AnilloTOTALp]
(* 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["TERMINACIONES DE FIBRA"]

```
Print["costeTerminacionesFibrax"]
Do[If[transtechx[[ix]]==Urbana||transtechx[[ix]]==Enterrado,
    costeTerminacionesFibrax[[ix]]=costeTerminacionesFibrax[[ix]]+2*TxFibraCosteTerminacion,0]
,{ix,nx}];
Print[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["DESAGREGACIÓN DEL TAMAÑO DE LOS ANILLOS: POR ARISTA"]

PRINT["A NIVEL PROVINCIAL"]

```
Print["disthostx modificado"]
Do[If[esAnillopx[[ix]]==1,
    If[e1Anillop[[provx[[ix]]]]≠ 0,
        disthostx[[ix]]=TamAnillop[[provx[[ix]]]]*e1remotehostTOTALx[[ix]]/e1AnilloTOTALp[[provx[[ix]]]]];
,{ix,nx}];
Print[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["A NIVEL DEPARTAMENTAL"]

```
Print["disttandemx modificado"]
Do[If[esAnillodx[[AnilloCabeceras[[ix]]]]!=NA,
    If[e1Anillod[[deptx[[AnilloCabeceras[[ix]]]]]]≠ 0,
        disttandemx[[AnilloCabeceras[[ix]]]]=TamAnillod[[deptx[[AnilloCabeceras[[ix]]]]]]*

e1hosttandemTOTALx[[AnilloCabeceras[[ix]]]]/e1AnilloTOTALd[[deptx[[AnilloCabeceras[[ix]]]]]]];,{ix,np}}
Print[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["ESTIMACIÓN DE INVERSIONES A NIVEL SATELITAL"]

PRINT["A NIVEL DE EQUIPOS EN LOS LUGARES REMOTOS"]

```
Print["satinvadslx"]
satinvadslx=Table[If[(sat1x[[ix]]==0)|| (e1hosttandemTOTALx[[ix]]==0),0,
                    (satantena+sateqtx)*(e1hosttandemadslx[[ix]]/e1hosttandemTOTALx[[ix]])],{ix,nx}];
Print[satinvadslx]
(* 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 ADSL, 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 ADSL entre el total de E1s a
nivel HT que incluye todos los servicios *)

Print["satinvTOTALx"]
satinvTOTALx=Table[If[(sat1x[[ix]]==0)|| (e1hosttandemTOTALx[[ix]]==0),0,(satantena+sateqtx)],{ix,nx}];
Print[satinvTOTALx]
(* Para todas las centrales que utilizan Tx satelital, se estima la inversión total (incluye todos los
servicios) en los equipos de Tx y la antenna satelital. *)

Print["satinvadsld"]
satinvadsld=zerod;
Do[satinvadsld[[deptx[[ix]]]]=satinvadsld[[deptx[[ix]]]]+satinvadslx[[ix]];;,{ix,nx}];
Print[satinvadsld]
(* Acumula a nivel departamental la variable satinvadslx: total de inversión en equipos de Tx y antenas
satelitales del servicio ADSL *)

Print["satinvTOTALd"]
satinvTOTALd=zerod;
Do[satinvTOTALd[[deptx[[ix]]]]=satinvTOTALd[[deptx[[ix]]]]+satinvTOTALx[[ix]];;,{ix,nx}];
Print[satinvTOTALd]
(* Acumula a nivel departamental la variable satinvTOTALx: total de inversión en equipos de Tx y antenas
satelitales de todos los servicios *)

Print["satinvads1"]
satinvads1=Plus@@satinvadsld;
Print[satinvads1]
(*Sumatoria a nivel nacional del total de inversión en equipos de Tx y antenas satelitales del servicio de
ADSL*)

Print["satinvTOTAL"]
satinvTOTAL=Plus@@satinvTOTALx;
Print[satinvTOTAL]
(* Sumatoria a nivel nacional del total de inversión en equipos de Tx y antenas satelitales considerando
todos los servicios *)
```

PRINT["ESTIMACIÓN DE INVERSIONES A NIVEL SATELITAL"]

PRINT["A NIVEL DE EQUIPOS EN LIMA"]

```
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["satlimainvadslx"]
satlimainvadslx=zerod;
Do[If[numerosat==0,satlimainvadslx[[ix]]=0,
satlimainvadslx[[ix]]=(satantena/numerosat+sateqtx)*(satinvads1/satinvTOTAL)*sat1x[[ix]]],{ix,nx}];
Print[satlimainvadslx]
(* Dedido a que todas las transmisiones satelitales vienen hasta Lima, en Lima se requiere también un equipo
de Tx y una antenna 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 antenna. 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["satlimainvTOTALx"]
satlimainvTOTALx=zerod;
Do[If[numerosat==0,satlimainvTOTALx[[ix]]=0,
satlimainvTOTALx[[ix]]=(satantena/numerosat+sateqtx)*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 antenna 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 antenna. *)

Print["satlimainvadsld"]
satlimainvadsld=zerod;
Do[satlimainvadsld[[deptx[[ix]]]]=satlimainvadsld[[deptx[[ix]]]]+satlimainvadslx[[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["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["satlimainvads1"]
satlimainvads1=Plus@@satlimainvadsld;
Print[satlimainvads1]
(* Acumula a nivel nacional la variable satlimainvadsld 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["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["A NIVEL DE ANCHO DE BANDA SATELITAL"]

(* E1s del servicio de Voz ya existe y está calculado previamente *)

```

Print["e1satalqxA1"]
e1satalqxA1=zerox;
Do[e1satalqxA1[[inx]]=e1hosttandemalqxA1[[inx]]*host1x[[inx]]*sat1x[[inx]]*(1-tandem1x[[inx]]),{inx,nx}];
Print[e1satalqxA1]
(* 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
e1hosttandemalqxA1 equivalente a la carga en E1s del servicio Circuitos Rango A *)

Print["e1satalqxB1"]
e1satalqxB1=zerox;
Do[e1satalqxB1[[inx]]=e1hosttandemalqxB1[[inx]]*host1x[[inx]]*sat1x[[inx]]*(1-tandem1x[[inx]]),{inx,nx}];
Print[e1satalqxB1]
(* 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
e1hosttandemalqxB1 equivalente a la carga en E1s del servicio Circuitos Rango B *)

Print["e1satalqxC1"]
e1satalqxC1=zerox;
Do[e1satalqxC1[[inx]]=e1hosttandemalqxC1[[inx]]*host1x[[inx]]*sat1x[[inx]]*(1-tandem1x[[inx]]),{inx,nx}];
Print[e1satalqxC1]
(* 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
e1hosttandemalqxC1 equivalente a la carga en E1s del servicio Circuitos Rango C *)

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+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
e1hostttandemTOTALx equivalente a la carga en E1s de todos los servicios *)

Print["Extrasatcostadslx"]
Extrasatcostadslx=2*Ceiling[e1satadslx*compresDCME]*numMHzporMb*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["ExtrasatcostTOTALx"]
ExtrasatcostTOTALx=2*Ceiling[e1satTOTALx*compresDCME]*numMHzporMb*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["Extrasatcostadslld"]
Extrasatcostadslld=zerod;
Do[Extrasatcostadslld[[deptx[[ix]]]]=Extrasatcostadslld[[deptx[[ix]]]]+Extrasatcostadslx[[ix]],{ix,nx}]
Print[Extrasatcostadslld]
(* Acumula por departamento la variable Extrasatcostadslx, Servicio de ADSL *)

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["ESTIMACIÓN DE COMPOSICION DE E1s (si es del Anillo o de tramo
independiente)"]

PRINT["A NIVEL HOST TANDEM"]

Print["e1compadslHTx"]
e1compadslHTx=zerox;
Do[If[esAnillodx[[ix]]#
NA,e1compadslHTx[[ix]]=Round[e1Anilloadslld[[deptx[[ix]]]]],e1compadslHTx[[ix]]=Round[e1hostttandemadslx[[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 e1Anilloadslld y si no es anillo simplemente la
variable e1hostttandemadslx, ambas referidas de manera exclusiva al servicio elegido *)

Print["e1compHTTOTALx"]
e1compHTTOTALx=zerox;
Do[If[esAnillodx[[ix]]#
NA,e1compHTTOTALx[[ix]]=e1AnilloTOTALd[[deptx[[ix]]]],e1compHTTOTALx[[ix]]=e1hostttandemTOTALx[[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 e1hostttandemTOTALx *)
```

```
PRINT["ESTIMACIÓN DE COMPOSICION DE E1s"]

PRINT["A NIVEL REMOTE HOST"]

Print["E1s por Servicio"];
e1compadslRHx=zerox;
e1compRHTOTALx=zerox;
Do[If[esAnillopx[[ix]]==1,
    e1compadslRHx[[ix]]=e1Anilloadslp[[provx[[ix]]]];
    e1compRHTOTALx[[ix]]=e1AnilloTOTALp[[provx[[ix]]]];
    e1compadslRHx[[ix]]=e1remotehostadslx[[ix]];
    e1compRHTOTALx[[ix]]=e1remotehostTOTALx[[ix]]];,{ix,nx}];
(* Especifica el número de E1s relevante para el servicio de ADSL y Totales: Vale para radio y fibra. Si la
central forma parte de un anillo departamental se tomar las variables e1Anilloadslp o e1AnilloTOTALp y si no
es anillo simplemente la variable e1remotehostadslx o e1remotehostTOTALx, ambas referidas de manera
exclusiva al servicio respectivo *)
Print[TableForm[{e1compadslRHx,e1compRHTOTALx},TableAlignments->{Center,Center},TableDirections->
{>{Row,Column},TableSpacing->{0,0},TableHeadings->{{"ADSL","Total"},Table[ix,{ix,nx}]}}]];
```

ESTIMACION DE INVERSIONES DE RADIO

PRINT["ESTIMACIÓN DE INVERSIONES EN RADIO A NIVEL CABECERA-TANDEM"]

```
Print["hophosttandemx"]
hophosttandemx=zerox;
Do[hophosttandemx[[ix]]=If[Ceiling[disttandemx[[ix]]/hop]>1,Ceiling[disttandemx[[ix]]/hop]-1,0];,{ix,nx}];
Print[hophosttandemx]
(* 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 *)

Print["numtributariosRHTx"]
numtributariosRHTx=Ceiling[e1hosttandemTOTALx/21];
Print[numtributariosRHTx]
(* Identifica el número de tributarios por central *)

Print["VelocidadRHTx"];
VelocidadRHTx=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,
    Do[If[e1compHTTOTALx[[ix]]>TxRadioLimites[[i]],VelocidadRHTx[[ix]]=i;
        ,{i,Length[TxRadioLimites]}];
    If[VelocidadRHTx[[ix]]<Length[TxRadioLimites],
        VelocidadRHTx[[ix]]=VelocidadRHTx[[ix]]+1];];
    ,{ix,nx}];
Print[VelocidadRHTx];
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)

Print["CosteFijoRHTx"]
CosteFijoRHTx=zerox;
Do[If[VelocidadRHTx[[ix]]!=0,
    CosteFijoRHTx[[ix]]=TxRadioCosteFijo[[VelocidadRHTx[[ix]]]];];,{ix,nx}];
Print[CosteFijoRHTx]
(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)

Print["CosteTributariosRHTx"]
CosteTributariosRHTx=zerox;
Do[If[VelocidadRHTx[[ix]]!=0,
    CosteTributariosRHTx[[ix]]=TxRadioCosteTributario[[VelocidadRHTx[[ix]]]];];,{ix,nx}];
Print[CosteTributariosRHTx]
(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

Print["CosteRepetidoresRHTx"]
CosteRepetidoresRHTx=zerox;
Do[If[VelocidadRHTx[[ix]]!=0,
    CosteRepetidoresRHTx[[ix]]=TxRadioCosteRep[[VelocidadRHTx[[ix]]]]*hophosttandemx[[ix]]];];
    ,{ix,nx}];
Print[CosteRepetidoresRHTx]
(* Especifica el costo del equipo repetidor asociado a la velocidad o capacidad requerida*)

Print["radioinvhosttandemadslx"]
radioinvhosttandemadslx=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,radioinvhosttandemadslx[[ix]]=(2*CosteFijoRHTx[[ix]]+CosteRepetidoresRHTx[[ix]]
+2*CosteTributariosRHTx[[ix]]* numtributariosRHTx[[ix]])*(e1compadslHTx[[ix]]/e1compHTTOTALx[[ix]]);
If[tandem1x[[ix]]==1,radioinvhosttandemadslx[[ix]]=0;];];,{ix,nx}];
Print[radioinvhosttandemadslx]
(* Estima el nivel de inversión Total e imputable al servicio de ADSL. Para ello, utiliza como factor o proporción imputable el ratio entre las variables e1compadslHTx (Especifica el número de E1s relevante para el servicio de ADSL) 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["radioinvhosttandemTOTALx"]
radioinvhosttandemTOTALx=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,radioinvhosttandemTOTALx[[ix]]=(2*CosteFijoRHTx[[ix]]+CosteRepetidoresRHTx[[ix]]
]+2*CosteTributariosRHTx[[ix]]* numtributariosRHTx[[ix]]);
If[tandem1x[[ix]]==1,radioinvhosttandemTOTALx[[ix]]=0;];];,{ix,nx}];
Print[radioinvhosttandemTOTALx]
(* 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 *)
```

PRINT["ESTIMACIÓN DE INVERSIONES EN RADIO A NIVEL REMOTA-CABECERA"]

```
Print["hopremotehostx"]
hopremotehostx=zerox;
Do[hopremotehostx[[ix]]=If[Ceiling[disthostx[[ix]]/hop]>1,Ceiling[disthostx[[ix]]/hop]-1,0];,{ix,nx}];
Print[hopremotehostx]
(* 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 *)

Print["numtributariosRRHx"]
numtributariosRRHx=Ceiling[e1remotehostx/21];
Print[numtributariosRRHx]
(* Identifica el número de tributarios por central *)

Print["VelocidadRRHx"];
VelocidadRRHx=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];];
    ,{ix,nx}];
Print[VelocidadRRHx];
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)

Print["CosteFijoRRHx"]
CosteFijoRRHx=zerox;
Do[If[VelocidadRRHx[[ix]]!=0,
    CosteFijoRRHx[[ix]]=TxRadioCosteFijo[[VelocidadRRHx[[ix]]]];];,{ix,nx}];
Print[CosteFijoRRHx]
(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)

Print["CosteTributariosRRHx"]
CosteTributariosRRHx=zerox;
Do[If[VelocidadRRHx[[ix]]!=0,
    CosteTributariosRRHx[[ix]]=TxRadioCosteTributario[[VelocidadRRHx[[ix]]]];];,{ix,nx}];
Print[CosteTributariosRRHx]
(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

Print["CosteRepetidoresRRHx"]
CosteRepetidoresRRHx=zerox;
Do[If[VelocidadRRHx[[ix]]!=0,
    CosteRepetidoresRRHx[[ix]]=TxRadioCosteRep[[VelocidadRRHx[[ix]]]]*hopremotehostx[[ix]];];
    ,{ix,nx}];
Print[CosteRepetidoresRRHx]
(* Especifica el costo del equipo repetidor asociado a la velocidad o capacidad requerida*)

Print["radioinvremotehostadslx"]
radioinvremotehostadslx=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,radioinvremotehostadslx[[ix]]=(2*CosteFijoRRHx[[ix]]+CosteRepetidoresRRHx[[ix]]
+2*CosteTributariosRRHx[[ix]]* numtributariosRRHx[[ix]])*(e1compadslRHx[[ix]]/e1compRHTOTALx[[ix]]);
If[tandem1x[[ix]]==1,radioinvremotehostadslx[[ix]]=0];];,{ix,nx}];
Print[radioinvremotehostadslx]
(* Estima el nivel de inversión Total e imputable al servicio de ADSL. Para ello, utiliza como factor o proporción imputable el ratio entre las variables e1compadslRHx (Especifica el número de E1s relevante para el servicio de ADSL) y la variable e1compRHTOTALx (Especifica el número de E1s 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["radioinvremotehostTOTALx"]
radioinvremotehostTOTALx=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
radioinvremotehostTOTALx[[ix]]=(2*CosteFijoRRHx[[ix]]+CosteRepetidoresRRHx[[ix]]+2*CosteTributariosRRHx[[ix]]
* numtributariosRRHx[[ix]]);
If[tandem1x[[ix]]==1,radioinvremotehostTOTALx[[ix]]=0];];,{ix,nx}];
Print[radioinvremotehostTOTALx]
(* 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) *)
```

CALCULO DE INVERSIÓN EN FIBRA

PRINT["CALCULO DEL COSTO POR KM DE FIBRA"]

```
Print["TxFibraCosteKmUrban: Fibra Canalizada"]
TxFibraCosteKmUrban=compfiburb*(TxFibraCosteEmpalme(1/TxFibraDistEntreEmpalmesKm)+
TxFibraCosteCanalizacionKm+TxFibraCosteCamara (1/TxFibraDistEntreCamarasKm));
Print[TxFibraCosteKmUrban]

Print["TxFibraCosteKmEnterrado: Fibra Enterrada"]
TxFibraCosteKmEnterrado=TxFibraCosteEmpalme(1/TxFibraDistEntreEmpalmesKm)+TxFibraCosteTrituboKm;
Print[TxFibraCosteKmEnterrado]

Print["TxFibraCosteKmAereo: Fibra Aerea"]
TxFibraCosteKmAereo=TxFibraCosteKmEnterrado;
Print[TxFibraCosteKmAereo]
```


PRINT["ESTIMACIÓN DE INVERSIONES EN FIBRA A NIVEL CABECERA-TANDEM"]

```

Print["Número Repetidoras de Fibra"]
repfibrax=Floor[repfibrax/hopfibra];
Print[repfibrax]
(* Especifica el número de repetidoras de Fibra por tramo *)

Print["numtributariosFHTx"]
numtributariosFHTx=Ceiling[e1hostttandemTOTALx/21];
Print[numtributariosFHTx]
(* Identifica el número de tributarios por central *)

Print["VelocidadFHTx"];
VelocidadFHTx=zeror;
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];];
  ,{ix,nx}];
Print[VelocidadFHTx];
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)

Print["CostePorKmFHTx"]
CostePorKmFHTx=zeror;
Do[If[transtechx[[ix]]==Urbana,CostePorKmFHTx[[ix]]=TxFibraCosteKmUrban;,
  If[transtechx[[ix]]==Enterrado,CostePorKmFHTx[[ix]]=TxFibraCosteKmEnterrado;,
    CostePorKmFHTx[[ix]]=TxFibraCosteKmAereo;];];,{ix,nx}];
Print[CostePorKmFHTx]
(* Especifica el costo asociado a la obra civil dependiendo del tipo de Fibra *)

Print["CosteFijoFHTx"]
CosteFijoFHTx=zeror;
Do[If[VelocidadFHTx[[ix]]!=0,
  CosteFijoFHTx[[ix]]=TxFibraCosteFijo[[VelocidadFHTx[[ix]]]];];,{ix,nx}];
Print[CosteFijoFHTx]
(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)

Print["CosteTributariosFHTx"]
CosteTributariosFHTx=zeror;
Do[If[VelocidadFHTx[[ix]]!=0,
  CosteTributariosFHTx[[ix]]=TxFibraCosteTributario[[VelocidadFHTx[[ix]]]];];,{ix,nx}];
Print[CosteTributariosFHTx]
(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

Print["CosteExtraFHTx"]
CosteExtraFHTx=zeror;
Do[If[esAnilloTx[[AnilloCabeceras[[ix]]]]!=NA,
  If[e1compHTTOTALx[[AnilloCabeceras[[ix]]]]!= 0,
    CosteExtraFHTx[[AnilloCabeceras[[ix]]]]=(CosteFijoFHTx[[AnilloCabeceras[[ix]]]]+
CosteTributariosFHTx[[AnilloCabeceras[[ix]]]]*Ceiling[e1compHTTOTALx[[AnilloCabeceras[[ix]]]]/21])*
e1hostttandemTOTALx[[AnilloCabeceras[[ix]]]]/e1compHTTOTALx[[AnilloCabeceras[[ix]]]]];];,{ix,np}]
Print[CosteExtraFHTx]
(* A nivel HT todos los costos deben ser imputados a las remotas. 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
(e1hostttandemTOTALx) y la carga total del anillo (e1compHTTOTALx) *)

Print["fiberinvhostttandemadslx1"]
fiberinvhostttandemadslx1=zeror;
Do[If[e1compHTTOTALx[[ix]]!=0,
  fiberinvhostttandemadslx1[[ix]]=If[esAnilloTx[[ix]]!=NA,1,2]*CosteFijoFHTx[[ix]]+If[esAnilloTx[[ix]]!=NA,1,2]*
CosteTributariosFHTx[[ix]]*numtributariosFHTx[[ix]]+CosteFijoFHTx[[ix]]*repfibrax[[ix]]+CosteExtraFHTx[[ix]];
  fiberinvhostttandemadslx1[[ix]]=fiberinvhostttandemadslx1[[ix]]*(e1compadslHTx[[ix]]/e1compHTTOTALx[[ix]]);
  If[tandem1x[[ix]]==1,fiberinvhostttandemadslx1[[ix]]=0];];,{ix,nx}]
Print[fiberinvhostttandemadslx1]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable al
servicio de ADSL, para lo cual se multiplica la inversión total por la fracción de asignación establecida.
Finalmente, se asegura que a nivel de centrales Tandem el resultado debe ser siempre cero *)

Print["fiberinvhostttandemTOTALx1"]
fiberinvhostttandemTOTALx1=zeror;
Do[If[e1compHTTOTALx[[ix]]!=0,
  fiberinvhostttandemTOTALx1[[ix]]=If[esAnilloTx[[ix]]!=NA,1,2]*CosteFijoFHTx[[ix]]+If[esAnilloTx[[ix]]!=NA,1,2]*
*CosteTributariosFHTx[[ix]]*numtributariosFHTx[[ix]]+CosteFijoFHTx[[ix]]*repfibrax[[ix]]+CosteExtraFHTx[[ix]];
  fiberinvhostttandemTOTALx1[[ix]]=fiberinvhostttandemTOTALx1[[ix]];
  If[tandem1x[[ix]]==1,fiberinvhostttandemTOTALx1[[ix]]=0];];,{ix,nx}]
Print[fiberinvhostttandemTOTALx1]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable a todos
los servicios. Finalmente, se asegura que a nivel de centrales Tandem el resultado debe ser siempre cero *)

Print["fiberinvhostttandemadslx2"]

```

```

fiberinvhosttandemadslx2=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,
fiberinvhosttandemadslx2[[ix]]=If[transtechx[[ix]]==
Urbana,costedelafibraurbKm,TxFibraCosteCableKm]*disttandemx[[ix]]+ costeTerminacionesFibrax[[ix]];
fiberinvhosttandemadslx2[[ix]]=fiberinvhosttandemadslx2[[ix]]*(e1compadslHTx[[ix]]/e1compHTTOTALx[[ix]]);
If[tandem1x[[ix]]==1,fiberinvhosttandemadslx2[[ix]]=0];},{ix,nx}]
Print[fiberinvhosttandemadslx2]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable
al servicio de ADSL, para lo cual se multiplica la inversión total por la fracción de asignación
establecida. Finalmente, se asegura que a nivel de centrales Tandem el resultado debe ser siempre cero *)

Print["fiberinvhosttandemTOTALx2"]
fiberinvhosttandemTOTALx2=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,
fiberinvhosttandemTOTALx2[[ix]]=If[transtechx[[ix]]==
Urbana,costedelafibraurbKm,TxFibraCosteCableKm]*disttandemx[[ix]]+ costeTerminacionesFibrax[[ix]];
fiberinvhosttandemTOTALx2[[ix]]=fiberinvhosttandemTOTALx2[[ix]];
If[tandem1x[[ix]]==1,fiberinvhosttandemTOTALx2[[ix]]=0];},{ix,nx}]
Print[fiberinvhosttandemTOTALx2]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable
a todos los servicios. Finalmente, se asegura que a nivel de centrales Tandem el resultado debe ser siempre
cero *)

Print["fiberinvhosttandemadslx3"]
fiberinvhosttandemadslx3=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,
fiberinvhosttandemadslx3[[ix]]=CostePorKmFHTx[[ix]] disttandemx[[ix]];
fiberinvhosttandemadslx3[[ix]]=fiberinvhosttandemadslx3[[ix]]*(e1compadslHTx[[ix]]/e1compHTTOTALx[[ix]]);
If[tandem1x[[ix]]==1,fiberinvhosttandemadslx3[[ix]]=0];},{ix,nx}]
Print[fiberinvhosttandemadslx3]
(* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia). Luego se identifica el
costo imputable al servicio de ADSL, para lo cual se multiplica la inversión total por la fracción de
asignación establecida. Finalmente, se asegura que a nivel de centrales Tandem el resultado debe ser
siempre cero *)

Print["fiberinvhosttandemTOTALx3"]
fiberinvhosttandemTOTALx3=zerox;
Do[If[e1compHTTOTALx[[ix]]!=0,
fiberinvhosttandemTOTALx3[[ix]]=CostePorKmFHTx[[ix]] disttandemx[[ix]];
fiberinvhosttandemTOTALx3[[ix]]=fiberinvhosttandemTOTALx3[[ix]];
If[tandem1x[[ix]]==1,fiberinvhosttandemTOTALx3[[ix]]=0];},{ix,nx}]
Print[fiberinvhosttandemTOTALx3]
(* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia). Luego se identifica el
costo imputable a todos los servicios. Finalmente, se asegura que a nivel de centrales Tandem el resultado
debe ser siempre cero *)

Print["fiberinvhosttandemadslx"]
fiberinvhosttandemadslx=fiberinvhosttandemadslx1+fiberinvhosttandemadslx2+fiberinvhosttandemadslx3;
Print[fiberinvhosttandemadslx]

Print["fiberinvhosttandemTOTALx"]
fiberinvhosttandemTOTALx=fiberinvhosttandemTOTALx1+fiberinvhosttandemTOTALx2+fiberinvhosttandemTOTALx3;
Print[fiberinvhosttandemTOTALx]

```

PRINT["ESTIMACIÓN DE INVERSIONES EN FIBRA A NIVEL REMOTA-CABECERA"]

```

Print["numtributariosFRHx"]
numtributariosFRHx=Ceiling[e1remotehostTOTALx/21];
Print[numtributariosFRHx]
(* Identifica el número de tributarios por central *)

Print["VelocidadFRHx"];
VelocidadFRHx=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];];
,{ix,nx}];
Print[VelocidadFHTx];
(* Identifica para cada arista cual es la velocidad de Tx o capacidad requerida *)

Print["CostePorKmFRHx"]
CostePorKmFRHx=zerox;
Do[If[transtechx[[ix]]==Urbana,CostePorKmFRHx[[ix]]=TxFibraCosteKmUrban;,
If[transtechx[[ix]]==Enterrado,CostePorKmFRHx[[ix]]=TxFibraCosteKmEnterrado;,
CostePorKmFRHx[[ix]]=TxFibraCosteKmAereo;];},{ix,nx}];
Print[CostePorKmFRHx]
(* Especifica el costo asociado a la obra civil dependiendo del tipo de Fibra *)

Print["CosteFijoFRHx"]
CosteFijoFRHx=zerox;
Do[If[VelocidadFRHx[[ix]]!=0,
CosteFijoFRHx[[ix]]=TxFibraCosteFijo[[VelocidadFRHx[[ix]]]];},{ix,nx}];
Print[CosteFijoFRHx]

```

```

(* Especifica el costo del equipo de Tx asociado a la velocidad o capacidad requerida*)

Print["CosteTributariosFRHx"]
CosteTributariosFRHx=zerox;
Do[If[VelocidadFRHx[[ix]]!=0,
    CosteTributariosFRHx[[ix]]=TxFibraCosteTributario[[VelocidadFRHx[[ix]]]];],{ix,nx}];
Print[CosteTributariosFRHx]
(* Especifica el costo del tributario asociado a la velocidad o capacidad requerida*)

Print["CosteExtraFRHx"]
CosteExtraFRHx=zerox;
Do[If[esAnillopx[[ix]]==1,
    If[e1AnilloTOTALp[[provx[[ix]]]]!=0,
    CosteExtraFRHx[[ix]]=(CosteFijoFRHx[[ix]]+CosteTributariosFRHx[[ix]]*Ceiling[e1AnilloTOTALp[[provx[[ix]]]]/21
    ])* e1remotehostTOTALx[[ix]]/e1compRHTOTALx[[ix]]];],{ix,nx}]
Print[CosteExtraFRHx]
(* 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) *)

Print["fiberinvremotehostadslx1"]
fiberinvremotehostadslx1=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
    fiberinvremotehostadslx1[[ix]]=If[esAnillopx[[ix]]==1,1,2]*CosteFijoFRHx[[ix]]+If[esAnillopx[[ix]]==
    1,1,2]*CosteTributariosFRHx[[ix]]numtributariosFRHx[[ix]]+CosteFijoFRHx[[ix]]*repfibrax[[ix]]
    +CosteExtraFRHx[[ix]];
    fiberinvremotehostadslx1[[ix]]=fiberinvremotehostadslx1[[ix]]*(e1compadslRHx[[ix]]/e1compRHTOTALx[[ix]]);
    If[host1x[[ix]]==1,fiberinvremotehostadslx1[[ix]]=0];],{ix,nx}]
Print[fiberinvremotehostadslx1]
(* Se estiman los costos asociados a los equipos: Equipo de Tx, Tributarios, Repetidora y Costo Extra.
Note 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. Luego de sumar dichos costos, se identifica el costo imputable al servicio
ADSL, para lo cual se multiplica la inversión total por la fracción de asignación establecida. Finalmente,
se asegura que a nivel de centrales Host el resultado debe ser siempre cero *)

Print["fiberinvremotehostTOTALx1"]
fiberinvremotehostTOTALx1=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
    fiberinvremotehostTOTALx1[[ix]]=If[esAnillopx[[ix]]==1,1,2]*CosteFijoFRHx[[ix]]+If[esAnillopx[[ix]]==
    1,1,2]*CosteTributariosFRHx[[ix]]numtributariosFRHx[[ix]]+CosteFijoFRHx[[ix]]*repfibrax[[ix]]
    +CosteExtraFRHx[[ix]];
    fiberinvremotehostTOTALx1[[ix]]=fiberinvremotehostTOTALx1[[ix]];
    If[host1x[[ix]]==1,fiberinvremotehostTOTALx1[[ix]]=0];],{ix,nx}]
Print[fiberinvremotehostTOTALx1]
(* Se estiman los costos asociados a los equipos: Equipo de Tx, Tributarios, Repetidora y Costo Extra.
Note 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. Luego de sumar dichos costos, se identifica el costo imputable a todos los
servicios. Finalmente, se asegura que a nivel de centrales Host el resultado debe ser siempre cero *)

Print["fiberinvremotehostadslx2"]
fiberinvremotehostadslx2=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
    fiberinvremotehostadslx2[[ix]]=If[transtechx[[ix]]==
    Urbana,costedelafibraurbKm,TxFibraCosteCableKm]*disthostx[[ix]]+ costeTerminacionesFibrax[[ix]];
    fiberinvremotehostadslx2[[ix]]=fiberinvremotehostadslx2[[ix]]*(e1compadslRHx[[ix]]/e1compRHTOTALx[[ix]]);
    If[host1x[[ix]]==1,fiberinvremotehostadslx2[[ix]]=0];],{ix,nx}]
Print[fiberinvremotehostadslx2]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable
al servicio de ADSL, para lo cual se multiplica la inversión total por la fracción de asignación
establecida. Finalmente, se asegura que a nivel de centrales Host el resultado debe ser siempre cero *)

Print["fiberinvremotehostTOTALx2"]
fiberinvremotehostTOTALx2=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
    fiberinvremotehostTOTALx2[[ix]]=If[transtechx[[ix]]==
    Urbana,costedelafibraurbKm,TxFibraCosteCableKm]*disthostx[[ix]]+ costeTerminacionesFibrax[[ix]];
    fiberinvremotehostTOTALx2[[ix]]=fiberinvremotehostTOTALx2[[ix]];
    If[host1x[[ix]]==1,fiberinvremotehostTOTALx2[[ix]]=0];],{ix,nx}]
Print[fiberinvremotehostTOTALx2]
(* 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. Luego de sumar dichos costos, se identifica el costo imputable
a todos los servicios. Finalmente, se asegura que a nivel de centrales Host el resultado debe ser siempre
cero *)

Print["fiberinvremotehostadslx3"]
fiberinvremotehostadslx3=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
    fiberinvremotehostadslx3[[ix]]=CostePorKmFRHx[[ix]]*disthostx[[ix]];
    fiberinvremotehostadslx3[[ix]]=fiberinvremotehostadslx3[[ix]]*(e1compadslRHx[[ix]]/e1compRHTOTALx[[ix]]);
    If[host1x[[ix]]==1,fiberinvremotehostadslx3[[ix]]=0];],{ix,nx}]
Print[fiberinvremotehostadslx3]
(* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia). Luego se identifica el
costo imputable al servicio de ADSL, para lo cual se multiplica la inversión total por la fracción de
asignación establecida. Finalmente, se asegura que a nivel de centrales Host el resultado debe ser siempre

```



```
cero *)

Print["fiberinvremotehostTOTALx3"]
fiberinvremotehostTOTALx3=zerox;
Do[If[e1compRHTOTALx[[ix]]!=0,
fiberinvremotehostTOTALx3[[ix]]=CostePorKmFRHx[[ix]]*disthostx[[ix]];
fiberinvremotehostTOTALx3[[ix]]=fiberinvremotehostTOTALx3[[ix]];
If[host1x[[ix]]==1,fiberinvremotehostTOTALx3[[ix]]=0];},{ix,nx}]
Print[fiberinvremotehostTOTALx3]
(* Se estiman los costos asociados a las obras civiles (Costo por Km * Distancia). Luego se identifica el
costo imputable a todos los servicios. Finalmente, se asegura que a nivel de centrales Host el resultado
debe ser siempre cero *)

Print["fiberinvremotehostadslx"]
fiberinvremotehostadslx=fiberinvremotehostadslx1+fiberinvremotehostadslx2+fiberinvremotehostadslx3;
Print[fiberinvremotehostadslx]

Print["fiberinvremotehostTOTALx"]
fiberinvremotehostTOTALx=fiberinvremotehostTOTALx1+fiberinvremotehostTOTALx2+fiberinvremotehostTOTALx3;
Print[fiberinvremotehostTOTALx]
```

INVERSIÓN COMPILADA

PRINT["INVERSIÓN COMPILADA A NIVEL CABECERA-TANDEM"]

```
Print["CosteAñadidoTrxPorDistanciaadslHTx"]
CosteAñadidoTrxPorDistanciaadslHTx=zerox;
Do[If[deptx[[ix]]!= LIMA && host1x[[ix]]==0,
CosteAñadidoTrxPorDistanciaadslHTx[[ix]]=AñadidoTrxPorKm*distance[ix,CentralRef];
CosteAñadidoTrxPorDistanciaadslHTx[[ix]]=CosteAñadidoTrxPorDistanciaadslHTx[[ix]]*
If[e1compHTTOTALx[[ix]]!=0,(e1compadslHTx[[ix]]/e1compHTTOTALx[[ix]]),0];},{ix,nx}];
Print[CosteAñadidoTrxPorDistanciaadslHTx]
(* 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 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 remotas
e1compHTTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel HT todo es cero *)

Print["CosteAñadidoTrxPorDistanciaHTTOTALx"]
CosteAñadidoTrxPorDistanciaHTTOTALx=zerox;
Do[If[deptx[[ix]]!= LIMA && host1x[[ix]]==0,
CosteAñadidoTrxPorDistanciaHTTOTALx[[ix]]=AñadidoTrxPorKm *distance[ix,CentralRef];
CosteAñadidoTrxPorDistanciaHTTOTALx[[ix]]=CosteAñadidoTrxPorDistanciaHTTOTALx[[ix]]*
If[e1compHTTOTALx[[ix]]!=0,1,0];},{ix,nx}];
Print[CosteAñadidoTrxPorDistanciaHTTOTALx]
(* 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. 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
e1compHTTOTALx=0, entonces costo añadido será cero. Conclusión, a nivel HT todo es cero *)

Print["invhosttandemadslx1"]
invhosttandemadslx1=Table[If[satx[[tandemx[[ix]]]],0,radioinvhosttandemadslx[[ix]]],{ix,nx}];
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemadslx1[[ix]]=fiberinvhosttandemadslx1[[ix]],{ix,nx}]
Do[invhosttandemadslx1[[ix]]=invhosttandemadslx1[[ix]]+CosteAñadidoTrxPorDistanciaadslHTx[[ix]]];,{ix,nx}];
Print[invhosttandemadslx1]
(* Compila o resume el total de inversiones a nivel de equipos a nivel HT imputable al servicio de ADSL. 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 radioinvhosttandemadslx. Luego, si la
Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será
fiberinvhosttandemadslx1. 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["invhosttandemTOTALx1"]
invhosttandemTOTALx1=Table[If[satx[[tandemx[[ix]]]],0,radioinvhosttandemTOTALx[[ix]]],{ix,nx}];
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemTOTALx1[[ix]]=fiberinvhosttandemTOTALx1[[ix]],{ix,nx}]
Do[invhosttandemTOTALx1[[ix]]=invhosttandemTOTALx1[[ix]]+CosteAñadidoTrxPorDistanciaHTTOTALx[[ix]]];,{ix,nx}];
Print[invhosttandemTOTALx1]
(* 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. Luego, si la
Tandem es Fibra, el nivel de inversión válido para cada una de las centrales del departamento será
fiberinvhosttandemTOTALx1. 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["invhosttandemadslld1"]
invhosttandemadslld1=zerod;
Do[invhosttandemadslld1[[deptx[[ix]]]]=invhosttandemadslld1[[deptx[[ix]]]]+invhosttandemadslx1[[ix]],{ix,nx}]
Print[invhosttandemadslld1]
(* Acumula a nivel departamental la variable invhosttandemadslx1 imputable al servicio de ADSL *)

Print["invhosttandemTOTALd1"]
invhosttandemTOTALd1=zerod;
Do[invhosttandemTOTALd1[[deptx[[ix]]]]=invhosttandemTOTALd1[[deptx[[ix]]]]+invhosttandemTOTALx1[[ix]],{ix,nx}
]
Print[invhosttandemTOTALd1]
(* Acumula a nivel departamental la variable invhosttandemTOTALx1 imputable a todos los servicios *)
```

```

Print["invhosttandemadslx2"]
invhosttandemadslx2=zerox;
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemadslx2[[ix]]=fiberinvhosttandemadslx2[[ix]],{ix,nx}]
Print[invhosttandemadslx2]
(* 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 fiberinvhosttandemadslx2, en todos los demás casos es cero *)

Print["invhosttandemTOTALx2"]
invhosttandemTOTALx2=zerox;
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemTOTALx2[[ix]]=fiberinvhosttandemTOTALx2[[ix]],{ix,nx}]
Print[invhosttandemTOTALx2]
(* 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 fiberinvhosttandemTOTALx2, en todos los demás casos es cero *)

Print["invhosttandemadslld2"]
invhosttandemadslld2=zerod;
Do[invhosttandemadslld2[[deptx[[ix]]]]=invhosttandemadslld2[[deptx[[ix]]]]+invhosttandemadslx2[[ix]],{ix,nx}]
Print[invhosttandemadslld2]
(* Acumula a nivel departamental la variable invhosttandemadslx2 imputable al servicio de ADSL *)

Print["invhosttandemTOTALd2"]
invhosttandemTOTALd2=zerod;
Do[invhosttandemTOTALd2[[deptx[[ix]]]]=invhosttandemTOTALd2[[deptx[[ix]]]]+invhosttandemTOTALx2[[ix]],{ix,nx}
]
Print[invhosttandemTOTALd2]
(* Acumula a nivel departamental la variable invhosttandemTOTALx2 imputable a todos los servicios *)

Print["invhosttandemadslx3"]
invhosttandemadslx3=zerox;
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemadslx3[[ix]]=fiberinvhosttandemadslx3[[ix]],{ix,nx}];
Print[invhosttandemadslx3]
(* 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 fiberinvhosttandemadslx3, en todos los demás casos es cero *)

Print["invhosttandemTOTALx3"]
invhosttandemTOTALx3=zerox;
Do[If[fiberx[[tandemx[[ix]]]],invhosttandemTOTALx3[[ix]]=fiberinvhosttandemTOTALx3[[ix]],{ix,nx}];
Print[invhosttandemTOTALx3]
(* 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 fiberinvhosttandemTOTALx3, en todos los demás casos es cero *)

Print["invhosttandemadslld3"]
invhosttandemadslld3=zerod;
Do[invhosttandemadslld3[[deptx[[ix]]]]=invhosttandemadslld3[[deptx[[ix]]]]+invhosttandemadslx3[[ix]],{ix,nx}]
Print[invhosttandemadslld3]
(* Acumula a nivel departamental la variable invhosttandemadslx3 imputable al servicio de ADSL *)

Print["invhosttandemTOTALd3"]
invhosttandemTOTALd3=zerod;
Do[invhosttandemTOTALd3[[deptx[[ix]]]]=invhosttandemTOTALd3[[deptx[[ix]]]]+invhosttandemTOTALx3[[ix]],{ix,nx}
]
Print[invhosttandemTOTALd3]
(* Acumula a nivel departamental la variable invhosttandemTOTALx3 imputable a todos los servicios *)

Print["invhosttandemadslx"]
invhosttandemadslx=invhosttandemadslx1+invhosttandemadslx2+invhosttandemadslx3;
Print[invhosttandemadslx]
(* Total de inversiones en Tx a nivel HT imputable al servicio de ADSL *)

Print["invhosttandemTOTALx"]
invhosttandemTOTALx=invhosttandemTOTALx1+invhosttandemTOTALx2+invhosttandemTOTALx3;
Print[invhosttandemTOTALx]
(* Total de inversiones en Tx a nivel HT imputable a todos los servicios *)

Print["invhosttandemadslld"]
invhosttandemadslld=zerod;
Do[invhosttandemadslld[[deptx[[ix]]]]=invhosttandemadslld[[deptx[[ix]]]]+invhosttandemadslx[[ix]],{ix,nx}]
Print[invhosttandemadslld]
(* Total de inversiones en Tx a nivel HT por departamento imputable al servicio de ADSL *)

Print["invhosttandemTOTALd"]
invhosttandemTOTALd=zerod;
Do[invhosttandemTOTALd[[deptx[[ix]]]]=invhosttandemTOTALd[[deptx[[ix]]]]+invhosttandemTOTALx[[ix]],{ix,nx}]
Print[invhosttandemTOTALd]
(* Total de inversiones en Tx a nivel HT por departamento imputable a todos los servicios *)

Print["invhosttandemadsl"]
invhosttandemadsl=Plus@@invhosttandemadslld;
Print[invhosttandemadsl]
(* Total de inversiones en Tx a nivel HT a nivel nacional imputable al servicio de ADSL *)

Print["invhosttandemTOTAL"]
invhosttandemTOTAL=Plus@@invhosttandemTOTALd;
Print[invhosttandemTOTAL]
(* Total de inversiones en Tx a nivel HT a nivel nacional imputable a todos los servicios *)

```


PRINT["INVERSIÓN COMPILADA A NIVEL REMOTA-CABECERA"]

```
Print["CosteAñadidoTrxPorDistanciaaadsLRHx"]
CosteAñadidoTrxPorDistanciaaadsLRHx=zeror;
Do[If[depts[[ix]]≠ LIMA && host1x[[ix]]==1,
  CosteAñadidoTrxPorDistanciaaadsLRHx[[ix]]=AñadidoTrxPorKm*distance[ix,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]]==1,
  CosteAñadidoTrxPorDistanciaRHTOTALx[[ix]]=AñadidoTrxPorKm*distance[ix,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["invremotehostadslx1"]
invremotehostadslx1=radioinvremotehostadslx;
Do[If[fiberx[[ix]],invremotehostadslx1[[ix]]=fiberinvremotehostadslx1[[ix]]],{ix,nx}]
Do[invremotehostadslx1[[ix]]=invremotehostadslx1[[ix]]+CosteAñadidoTrxPorDistanciaaadsLRHx[[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["invremotehostadsl1"]
invremotehostadsl1=zerod;
Do[invremotehostadsl1[[depts[[ix]]]]=invremotehostadsl1[[depts[[ix]]]]+invremotehostadslx1[[ix]],{ix,nx}]
Print[invremotehostadsl1]
(* Acumula a nivel departamental la variable invremotehostadslx1 imputable al servicio de ADSL *)

Print["invremotehostTOTALd1"]
invremotehostTOTALd1=zerod;
Do[invremotehostTOTALd1[[depts[[ix]]]]=invremotehostTOTALd1[[depts[[ix]]]]+invremotehostTOTALx1[[ix]],{ix,nx}]
Print[invremotehostTOTALd1]
(* Acumula a nivel departamental la variable invremotehostTOTALx1 imputable a todos los servicios *)

Print["invremotehostadslx2"]
invremotehostadslx2=zeror;
Do[If[fiberx[[ix]],invremotehostadslx2[[ix]]=fiberinvremotehostadslx2[[ix]]],{ix,nx}]
Do[If[depts[[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=zeror;
Do[If[fiberx[[ix]],invremotehostTOTALx2[[ix]]=fiberinvremotehostTOTALx2[[ix]]],{ix,nx}]
Do[If[depts[[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["invremotehostadsld2"]
invremotehostadsld2=zerod;
Do[invremotehostadsld2[[deptx[[ix]]]]=invremotehostadsld2[[deptx[[ix]]]]+invremotehostadsldx2[[ix]],{ix,nx}]
Print[invremotehostadsld2]
(* Acumula a nivel departamental la variable invremotehostadsldx2 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["invremotehostadsldx3"]
invremotehostadsldx3=zerod;
Do[If[fiberx[[ix]],invremotehostadsldx3[[ix]]=fiberinvremotehostadsldx3[[ix]],{ix,nx}]
Do[If[deptx[[ix]]!=LIMA && transtechx[[ix]]==Enterrado,
  invremotehostadsldx3[[ix]]=0.55*invremotehostadsldx3[[ix]]];,{ix,nx}];
Print[invremotehostadsldx3]
(* 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 fiberinvremotehostadsldx3, 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["invremotehostadsld3"]
invremotehostadsld3=zerod;
Do[invremotehostadsld3[[deptx[[ix]]]]=invremotehostadsld3[[deptx[[ix]]]]+invremotehostadsldx3[[ix]],{ix,nx}]
Print[invremotehostadsld3]
(* Acumula a nivel departamental la variable invremotehostadsldx3 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["invremotehostdsldx"]
invremotehostdsldx=invremotehostadsldx1+invremotehostadsldx2+invremotehostadsldx3;
Print[invremotehostdsldx]
(* 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["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["invremotehostadsld"]
invremotehostadsld=Plus@@invremotehostadsld;
Print[invremotehostadsld]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable al servicio de ADSL *)

Print["invremotehostTOTAL"]
invremotehostTOTAL=Plus@@invremotehostTOTALd;
Print[invremotehostTOTAL]
(* Total de inversiones en Tx a nivel RH a nivel nacional imputable a todos los servicios *)

```

PRINT["RESUMEN DE INVERSIONES POR ELEMENTO Y DEPARTAMENTO"]

PRINT["A NIVEL DE DEPARTAMENTO"]

```
Print["InvRemotaHostd"];
InvRemotaHostadsld=invremotehostadsld;
InvRemotaHostTOTALd=invremotehostTOTALd;
Print[MatrixForm[{InvRemotaHostadsld,InvRemotaHostTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]];
(* Inversion total en Transmisión a nivel RH por departamento imputable al servicio de ADSL *)
(* Inversion total en Transmisión a nivel RH por departamento imputable a todos los servicios *)

Print["InvHostTandemd"];
InvHostTandemadsld=invhosttandemadsld;
InvHostTandemTOTALd=invhosttandemTOTALd;
Print[MatrixForm[{InvHostTandemadsld,InvHostTandemTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]];
(* Inversion total en Transmisión a nivel HT por departamento imputable al servicio de ADSL *)
(* Inversion total en Transmisión a nivel HT por departamento imputable a todos los servicios *)
```

CALCULO DE FACTORES DE ANUALIZACION

PRINT["FACTORES DE ANUALIZACIÓN"]

```
OyMCx = networkopfrac + switchmaintfrac;
OyMTx = networkopfrac + transmaintfrac;
OyMSupp = networkopfrac + supportmaintfrac;
VuCx = Round[1 / switchdeprfrac];
VuTx1 = Round[1 / transdeprfrac];
VuTx2 = Round[1 / fibertransdeprfrac];
VuTx3 = Round[1 / petransdeprfrac];
VuSupp = Round[1 / supportdeprfrac];
VuEdif = Round[1 / edifsupportdeprfrac];

FactorCx = OyMCx + overheadfrac +  $\frac{retcap}{1 - (1 + retcap)^{-VuCx}}$  +
supportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuSupp}} \right)$  + edifsupportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuEdif}} \right)$ ;

FactorTx1 = OyMTx + overheadfrac +  $\frac{retcap}{1 - (1 + retcap)^{-VuTx1}}$  + supportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuSupp}} \right)$  +
edifsupportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuEdif}} \right)$ ;

FactorTx2 = OyMTx + overheadfrac +  $\frac{retcap}{1 - (1 + retcap)^{-VuTx2}}$  + supportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuSupp}} \right)$  +
edifsupportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuEdif}} \right)$ ;

FactorTx3 = OyMTx + overheadfrac +  $\frac{retcap}{1 - (1 + retcap)^{-VuTx3}}$  + supportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuSupp}} \right)$  +
edifsupportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuEdif}} \right)$ ;

FactorOyMTx = OyMTx + overheadfrac + supportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuSupp}} \right)$  +
edifsupportinvfrac  $\left( OyMSupp + \frac{retcap}{1 - (1 + retcap)^{-VuEdif}} \right)$ ;

Print["Factores Anuales"]
Print[MatrixForm[{FactorCx, FactorTx1, FactorTx2, FactorTx3, FactorOyMTx},
TableDirections->{Row, Column}, TableHeadings->{{"Cx", "Tx1", "Tx2", "Tx3", "OyMTx"}, Automatic}]];
```

PRINT["ESTIMACIÓN DE LOS NIVELES DE GASTO EN TX POR ELEMENTO Y POR CENTRAL"]

PRINT["A NIVEL REMOTE HOST"]

```
Print["CTTxRemotaHostadslx: Costo Total de Tx RemotaHost por Central imputable al servicio de ADSL"]
CTTxRemotaHostadslx=invremotehostadslx1*FactorTx1+invremotehostadslx2*FactorTx2+invremotehostadslx3*FactorTx3
;
Print[CTTxRemotaHostadslx]
(* Gasto Total en Transmisión a nivel RH por central imputable al servicio de ADSL *)

Print["CTTxRemotaHostTOTALx: Costo Total de Tx RemotaHost por Central imputable a todos los servicios"]
CTTxRemotaHostTOTALx=invremotehostTOTALx1*FactorTx1+invremotehostTOTALx2*FactorTx2+invremotehostTOTALx3*FactorTx3;
Print[CTTxRemotaHostTOTALx]
(* Gasto Total en Transmisión a nivel RH por central imputable a todos los servicios *)
```

PRINT["ESTIMACIÓN DE LOS NIVELES DE GASTO EN TX POR ELEMENTO Y POR CENTRAL"]

PRINT["A NIVEL HOST TANDEM"]

```
Print["CTTxHostTandemadslx: Costo Total de Tx HostTandem por Central imputable al servicio de ADSL"]
CTTxHostTandemadslx=invhosttandemadslx1*FactorTx1+invhosttandemadslx2*FactorTx2+invhosttandemadslx3*FactorTx3
+(FactorTx1*satinvadslx+Extrasatcostadslx)+(FactorTx1*satlimainvadslx+Extrasatcostadslx);
Print[CTTxHostTandemadslx]
(* Gasto Total en Transmisión a nivel HT por central imputable al servicio de ADSL. 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["CTTxHostTandemTOTALx: Costo Total de Tx HostTandem por Central imputable a todos los servicios"]
CTTxHostTandemTOTALx=invhosttandemTOTALx1*FactorTx1+invhosttandemTOTALx2*FactorTx2+invhosttandemTOTALx3*FactorTx3
+(FactorTx1*satinvTOTALx+ExtrasatcostTOTALx)+(FactorTx1*satlimainvTOTALx+ExtrasatcostTOTALx);
Print[CTTxHostTandemTOTALx]
(* Gasto Total en Transmisión a nivel HT por central 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["ESTIMACIÓN DE LOS NIVELES DE GASTO POR ELEMENTO"]

PRINT["A NIVEL DEPARTAMENTAL"]

Print["Costo Total de Tx RemotaHost por Dpto"]

CTTxRemotaHostadsld=invremotehostadsld1*FactorTx1+invremotehostadsld2*FactorTx2+invremotehostadsld3*FactorTx3;
;
CTTxRemotaHostTOTALd=invremotehostTOTALd1*FactorTx1+invremotehostTOTALd2*FactorTx2+invremotehostTOTALd3*FactorTx3;

Print[MatrixForm[{CTTxRemotaHostadsld,CTTxRemotaHostTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];

(* Gasto Total en Transmisión a nivel RH por departamento imputable al servicio de ADSL *)
(* Gasto Total en Transmisión a nivel RH por departamento imputable a todos los servicios *)

Print["Costo Total de Tx HostTandem por Dpto"]

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[MatrixForm[{CTTxHostTandemadsld,CTTxHostTandemTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];

(* Gasto Total en Transmisión a nivel HT por departamento imputable al servicio de ADSL. 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 *)
(* 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["Costo Total de Tx por Dpto"]

CTTxadsld=CTTxRemotaHostadsld+CTTxHostTandemadsld;
CTTxTOTALd=CTTxRemotaHostTOTALd+CTTxHostTandemTOTALd;
Print[MatrixForm[{CTTxadsld,CTTxTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];
(* Gasto Total en Transmisión por departamento imputable al servicio de ADSL *)
(* Gasto Total en Transmisión por departamento imputable a todos los servicios *)

PRINT["GASTOS FINALES A NIVEL NACIONAL"]

Print["Gasto Total en Transmisión a nivel nacional"]

CTTxadsl=Plus@@CTTxadsld;
CTTxTOTAL=Plus@@CTTxTOTALd;
Print[MatrixForm[{CTTxadsl,CTTxTOTAL},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];
(* Gasto Total en Transmisión a nivel nacional imputable al servicio de ADSL *)
(* Gasto Total en Transmisión a nivel nacional imputable a todos los servicios *)

PRINT["A NIVEL TOTAL"]

Print["Costo Total Anual de los Elementos de red por Dpto"]

CTadsld=CTTxRemotaHostadsld+CTTxHostTandemadsld;
CTTOTALd=CTTxRemotaHostTOTALd+CTTxHostTandemTOTALd;
Print[MatrixForm[{CTadsld,CTTOTALd},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];

(* Gasto Total por departamento imputable al servicio de ADSL. Incluye todos los elementos de red: centrales remotas, centrales cabeceras, centrales tandem, transmisión RH, transmisión HT y señalización *)
(* 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["Costo Total a nivel nacional"]

CTadsl=Plus@@CTadsld;
CTTOTAL=Plus@@CTTOTALd;
Print[MatrixForm[{CTadsl,CTTOTAL},TableDirections->{Row,Column},TableHeadings->{{"ADSL","TOTAL"},Automatic}]]];

(* Gasto Total a nivel nacional imputable al servicio de ADSL. Incluye los elementos de red: transmisión RH y transmisión HT *)
(* 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 *)

CÁLCULOS LDN

```
PRINT["ESTIMACIONES DE COSTOS DE LA RED INTERDEPRTAMENTAL: IDENTIFICACIÓN DE CARGAS POR PUNTO DENTRO DE LA MATRIZ DE ENCAMINAMIENTO "]
```

```
FactHC=0.002;
```

```
PRINT["LECTURA DE INFORMACIÓN DE TRÁFICO"]
```

```
Print["traficoLDS3origDest: Segundos LDN A FIJOS"]
```

```
OpenRead["input_traficoLDS3.txt"];
```

```
traficoLDS3origDest=ReadList["input_traficoLDS3.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS3origDest]]
```

(* Lee el tráfico en segundos de las llamadas de LDN de fijo a fijo: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental *)

```
Print["traficoLDS4origDest: Segundos LDI"]
```

```
OpenRead["input_traficoLDS4.txt"];
```

```
traficoLDS4origDest=ReadList["input_traficoLDS4.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS4origDest]]
```

(* Lee el tráfico en segundos de las llamadas de LDI: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. Dado que todo el tráfico LDI sale desde Lima, para cada departamento sólo aparece tráfico con destino hacia Lima, y cero para los demás destinos *)

```
Print["traficoLDS5origDest: Segundos LDN A MOVILES"]
```

```
OpenRead["input_traficoLDS5.txt"];
```

```
traficoLDS5origDest=ReadList["input_traficoLDS5.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS5origDest]]
```

(* Lee el tráfico en segundos de las llamadas de LDN de fijo a móvil: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. *)

```
Print["traficoLDS6origDest: Segundos de Llamadas Locales a fijos realizadas con Tarjeta 147"]
```

```
OpenRead["input_traficoLDS6.txt"];
```

```
traficoLDS6origDest=ReadList["input_traficoLDS6.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS6origDest]]
```

(* Lee el tráfico en segundos de las llamadas locales a fijos realizadas con tarjeta 147: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. Dado que todo local desde 147 debe venir primero a Lima (aquí está la plataforma), para cada departamento sólo aparece tráfico con destino hacia Lima, y cero para los demás destinos, y en la cado de Lima se aprecia tráfico hacia todos los departamentos *)

```
Print["traficoLDS7origDest: Segundos de Llamadas Locales a móviles realizadas con Tarjeta 147"]
```

```
OpenRead["input_traficoLDS7.txt"];
```

```
traficoLDS7origDest=ReadList["input_traficoLDS7.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS7origDest]]
```

(* Lee el tráfico en segundos de las llamadas locales a móviles realizadas con tarjeta 147: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental.*)

```
Print["traficoLDS8origDest: Segundos de Llamadas LDN a fijos realizadas con Tarjeta 147"]
```

```
OpenRead["input_traficoLDS8.txt"];
```

```
traficoLDS8origDest=ReadList["input_traficoLDS8.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS8origDest]]
```

(* Lee el tráfico en segundos de las llamadas LDN a fijos realizadas con tarjeta 147: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. Dado que todo local desde 147 debe venir primero a Lima (aquí está la plataforma), para cada departamento sólo aparece tráfico con destino hacia Lima, y cero para los demás destinos, y en la cado de Lima se aprecia tráfico hacia todos los departamentos *)

```
Print["traficoLDS9origDest: Segundos de Llamadas LDN a móviles realizadas con Tarjeta 147"]
```

```
OpenRead["input_traficoLDS9.txt"];
```

```
traficoLDS9origDest=ReadList["input_traficoLDS9.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS9origDest]]
```

(* Lee el tráfico en segundos de las llamadas LDN a móviles realizadas con tarjeta 147: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. *)

```
Print["traficoLDS10origDest: Segundos de Llamadas LDI realizadas con Tarjeta 147"]
```

```
OpenRead["input_traficoLDS10.txt"];
```

```
traficoLDS10origDest=ReadList["input_traficoLDS10.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS10origDest]]
```

(* Lee el tráfico en segundos de las llamadas LDI realizadas con tarjeta 147: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental. Dado que todo el tráfico desde 147 debe venir primero a Lima (aquí está la plataforma), para cada departamento sólo aparece tráfico con destino hacia Lima, y cero para los demás destinos. *)

```
Print["traficoLDS12origDest: Segundos de Otras Llamadas de LD "]
```

```
OpenRead["input_traficoLDS12.txt"];
```

```
traficoLDS12origDest=ReadList["input_traficoLDS12.txt",Real,RecordLists->True];
```

```
Print[MatrixForm[traficoLDS12origDest]]
```

(* Lee el tráfico en segundos de otras llamadas LD: Es una matriz de 24 por 24 que establece el tráfico de departamento a departamento (origen-destino). El orden de los departamentos es el mismo que el utilizado en el modelo intradepartamental.*)

```

Print["traficoLDSTOTALOrigDest: E1s TOTALES"];
traficoLDSTOTALOrigDest=Table[traficoLDS3OrigDest[[ix,jx]]+traficoLDS4OrigDest[[ix,jx]]+traficoLDS5OrigDest[[ix,jx]]+traficoLDS6OrigDest[[ix,jx]]+traficoLDS7OrigDest[[ix,jx]]+traficoLDS8OrigDest[[ix,jx]]+traficoLDS9OrigDest[[ix,jx]]+traficoLDS10OrigDest[[ix,jx]]+traficoLDS12OrigDest[[ix,jx]],{ix,nd},{jx,nd}];
Print[MatrixForm[traficoLDSTOTALOrigDest]];
(* Acumula el tráfico de todos los servicios en una sola matrix de 24 por 24, que determina el tráfico total por ruta *)

Print["ErlangsLD: Tráfico en Erlangs por ruta"];
ErlangsLD=Table[If[ix<=jx,FactHC*(traficoLDSTOTALOrigDest[[ix,jx]]+If[ix!=jx,traficoLDSTOTALOrigDest[[jx,ix]],0])/(60*60),0],{ix,nd},{jx,nd}];
Print[MatrixForm[ErlangsLD]];
(* Esta sección Convierte, los segundos mensuales en Erlangs, para cada ruta de manera independiente, y contando ya con los tráficos de todos los servicios. Trabaja agrupando las rutas entrantes y salientes *)

Print["traficoLDSOrigDest: E1s TOTALES"];
PctLince=0.58;
PctCercado=1-PctLince;
traficoLDSOrigDest=Table[If[(ix==LIMA||jx==LIMA)&&jx!=MADREDEDIOS&&ix!=MADREDEDIOS,

Ceiling[CalcTrunk[ErlangsLD[[ix,jx]]*PctLince,prob]/30/fiberfill]+Ceiling[CalcTrunk[ErlangsLD[[ix,jx]]*PctCercado,prob]/30/fiberfill],
Ceiling[CalcTrunk[ErlangsLD[[ix,jx]],prob]/30/fiberfill]],{ix,nd},{jx,nd}];
Print[MatrixForm[traficoLDSOrigDest]];
(* Esta tabla convierte los Erlangs a Circuitos por cada ruta. En este procedimiento se separa los circuitos que van al centro de conmutación nodal de LINCE con el de EL CERCADO, excepto el caso de Madre de Dios, que va solo a Lince, Luego convierte los circuitos por ruta a E1s por ruta *)

PRINT["LECTURA DE E1s de CIRCUITOS DE LD y ADSL"]

Print["traficoLDadslOrigDest: Demanda de E1s entre departamentos para red ADSL"]
OpenRead["input_traficoLDadsl.txt"];
traficoLDadslOrigDest=ReadList["input_traficoLDadsl.txt",Real,RecordLists->True];
Clear[traficoLDadslOrigDest];
traficoLDadslOrigDest=inputLDadsl;
Print[MatrixForm[traficoLDadslOrigDest]];
(* Matriz de 24 por 24 que especifica la carga entre departamentos por servicio de ADSL. Es tráfico de Tandem a Tandem. No se utiliza el dato en el archivo de texto, sino que se construye a partir de las demandas departamentales*)

Print["traficoAlqLDE1sOrigDestA1: Demanda de E1s entre departamentos para los circuitos tipo A"]
OpenRead["input_traficoAlqLDE1sA1.txt"];
traficoAlqLDE1sOrigDestA1=ReadList["input_traficoAlqLDE1sA1.txt",Real,RecordLists->True];
Print[MatrixForm[traficoAlqLDE1sOrigDestA1]];
(* Matriz de 24 por 24 que especifica la carga entre departamentos por los circuitos interdepartamentales del tipo A. Es tráfico de Tandem a Tandem. *)

Print["traficoAlqLDE1sOrigDestB1: Demanda de E1s entre departamentos para los circuitos tipo B"]
OpenRead["input_traficoAlqLDE1sB1.txt"];
traficoAlqLDE1sOrigDestB1=ReadList["input_traficoAlqLDE1sB1.txt",Real,RecordLists->True];
Print[MatrixForm[traficoAlqLDE1sOrigDestB1]];
(* Matriz de 24 por 24 que especifica la carga entre departamentos por los circuitos interdepartamentales del tipo B. Es tráfico de Tandem a Tandem. *)

Print["traficoAlqLDE1sOrigDestC1: Demanda de E1s entre departamentos para los circuitos tipo C"]
OpenRead["input_traficoAlqLDE1sC1.txt"];
traficoAlqLDE1sOrigDestC1=ReadList["input_traficoAlqLDE1sC1.txt",Real,RecordLists->True];
Print[MatrixForm[traficoAlqLDE1sOrigDestC1]];
(* Matriz de 24 por 24 que especifica la carga entre departamentos por los circuitos interdepartamentales del tipo C. Es tráfico de Tandem a Tandem. *)

Print["traficoAlqLDE1sOrigDestABC2: Demanda de E1s entre departamentos para otros circuitos tipo A, B y C"]
OpenRead["input_traficoAlqLDE1sABC2.txt"];
traficoAlqLDE1sOrigDestABC2=ReadList["input_traficoAlqLDE1sABC2.txt",Real,RecordLists->True];
Clear[traficoAlqLDE1sOrigDestABC2];
traficoAlqLDE1sOrigDestABC2=inputLDNucNuc;
Print[MatrixForm[traficoAlqLDE1sOrigDestABC2]];
TOTALE1ALQABC2=TOTALE1ALQABC2+Plus@@Plus@@traficoAlqLDE1sOrigDestABC2;
(* Matriz de 24 por 24 que especifica la carga entre departamentos por otros circuitos interdepartamentales del tipo A, B y C. Es tráfico de Tandem a Tandem. *)

Print["traficoAlqLDE1sOrigDestABCnop: Demanda de E1s entre departamentos para los circuitos de No Operadores"]
OpenRead["input_traficoAlqLDE1sABCnop.txt"];
traficoAlqLDE1sOrigDestABCnop=ReadList["input_traficoAlqLDE1sABCnop.txt",Real,RecordLists->True];
Print[MatrixForm[traficoAlqLDE1sOrigDestABCnop]];
(* Matriz de 24 por 24 que especifica la carga entre departamentos por los circuitos interdepartamentales de no operadores. Es tráfico de Tandem a Tandem. *)

PRINT["MATRIZ DE ENCAMINAMIENTO"]

calculaservicioLD=d[[2]];

Print["MATRIZ DE ENCAMINAMIENTO"]
OpenRead["input_encaminamientoLD.txt"];
encaminamientoLD=ReadList["input_encaminamientoLD.txt",Real,RecordLists->True];
Print[MatrixForm[encaminamientoLD]]

PRINT["ACUMULACIÓN DEL TRÁFICO EN SEGUNDOS POR PUNTO DENTRO DE LA MATRIZ DE ENCAMINAMIENTO"]

```

(* Para cada uno de los tipos de servicios: En el caso de los tráfico de voz se acumula por el momento el número de segundos, y en el caso de circuitos se acumulan directamente el número de E1s. Para cada tipo de servicio se hace lo siguiente: *)

(* Se crea una lista de 576 elementos, equivalente a 24*24. Cada resultado equivale al tráfico total que se acumula en el tramo respectivo, así, en los primeros 27 resultados tendremos los tráficos totales que se han acumulado en los tramos 1-2,1-3,1-4,1-5...1-24,2-1,2-2,2-3 y así sucesivamente hasta el total acumulado en el tramo 24-24.

Ejemplo: Estamos en el punto (2,3), en ese punto tenemos que el origen≠destino, y asumamos que el tráfico es de 3 segundos, es decir mayor a cero, con lo cual, cumplida las dos condiciones entramos a trabajar y acumular las cargas en los tramos por los cuales se pasa para llegar del departamento 2 al 3 (si no se cumplieran las condiciones pasaríamos a evaluar el punto (2,4).

Como primer paso se identifica en la matriz de encaminamiento el dato que aparece en el punto (2,3), el cual es igual a 5, lo cual quiere decir que debemos ir del departamento 2 al 5, con lo cual ahora debemos ir de 5 a 3, es decir, es como si estuviéramos en el punto (5,3); nuevamente, como origen≠destino buscamos en la matriz de encaminamiento el dato que aparece en el punto (5,3), el cual es igual a 8, lo cual quiere decir que debemos ir del departamento 5 al 8, es decir, es como si estuviéramos en el punto (8,3);nuevamente, como origen≠destino buscamos en la matriz de encaminamiento el dato que aparece en el punto (8,3), el cual es igual a 3, lo cual quiere decir que debemos ir del departamento 8 al 3, con lo cual llegamos al punto de destino deseado. Luego de todo ello, se empieza a trabajar a partir del punto (2,4) y así sucesivamente hasta llegar al punto (24,24). En el ejemplo, es claro que el tráfico de 3 segundos se debe considerar en los tramos 2-5,5-8, y 8-3. Obviamente todo el proceso arranca desde el punto (1,1), en el cual siendo origen=destino se pasa automáticamente al (1,2) y así hasta evaluar el punto (24,24). Nótese que por el momento, los tramos a-b y b-a son considerados como tramos distintos. *)

```
Print["traficoLDS: E1s Totales por Tramo para el tráfico de Voz LD"]
traficoLDS=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoLDSOrigDest[[ix]][[jx]]>0,
While[origen≠destino,
traficoLDS[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoLDS[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+traficoLDSOrigDest[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoLDS]
```

PRINT["ACUMULACIÓN DE E1s DE CIRCUITOS LD POR PUNTO DENTRO DE LA MATRIZ DE ENCAMINAMIENTO "]

```
Print["traficoLDadsl: E1s por el Servicio de ADSL"]
traficoLDadsl=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoLDadslOrigDest[[ix]][[jx]]>0,
While[origen≠destino,
traficoLDadsl[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoLDadsl[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+traficoLDadslOrigDest[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoLDadsl]
```

```
Print["traficoAlqLDE1sA1: E1s por los circuitos LD Rango A"]
traficoAlqLDE1sA1=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoAlqLDE1sOrigDestA1[[ix]][[jx]]>0,
While[origen≠destino,
traficoAlqLDE1sA1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoAlqLDE1sA1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+
traficoAlqLDE1sOrigDestA1[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoAlqLDE1sA1]
```

```
Print["traficoAlqLDE1sB1: E1s por los circuitos LD Rango B"]
traficoAlqLDE1sB1=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoAlqLDE1sOrigDestB1[[ix]][[jx]]>0,
While[origen≠destino,
traficoAlqLDE1sB1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoAlqLDE1sB1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+
traficoAlqLDE1sOrigDestB1[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoAlqLDE1sB1]
```

```
Print["traficoAlqLDE1sC1: E1s por los circuitos LD Rango C"]
traficoAlqLDE1sC1=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoAlqLDE1sOrigDestC1[[ix]][[jx]]>0,
While[origen≠destino,
traficoAlqLDE1sC1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoAlqLDE1sC1[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+
traficoAlqLDE1sOrigDestC1[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoAlqLDE1sC1]
```

```
Print["traficoAlqLDE1sABC2: E1s por otros circuitos LD Rango A, B y C"]
traficoAlqLDE1sABC2=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoAlqLDE1sOrigDestABC2[[ix]][[jx]]>0,
While[origen≠destino,
traficoAlqLDE1sABC2[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoAlqLDE1sABC2[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+
traficoAlqLDE1sOrigDestABC2[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoAlqLDE1sABC2]
```

```
Print["traficoAlqLDE1sABCnop: E1s por los circuitos LD de No Operadores"]
```



```

traficoAlqLDE1sABCnop=Table[0,{id,nd*nd}];
Do[Do[origen=ix;destino=jx;
If[jx≠ix&&traficoAlqLDE1sOrigDestABCnop[[ix]][[jx]]>0,
While[origen≠destino,
traficoAlqLDE1sABCnop[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]=
traficoAlqLDE1sABCnop[[nd*(origen-1)+encaminamientoLD[[origen]][[destino]]]+
traficoAlqLDE1sOrigDestABCnop[[ix]][[jx]];
origen=encaminamientoLD[[origen]][[destino]];];];,{jx,nd}];,{ix,nd}];
Print[traficoAlqLDE1sABCnop]

PRINT["ESTMACIÓN DEL COSTO DE LOS TRAMOS INTERDEPARTAMENTALES VÍA FIBRA (EN LA COSTA)"]

PRINT["ACUMULACIÓN DE E1s DE VOZ POR TRAMO DE FIBRA: RED DORSAL DE LA COSTA"]

DeptOrigenF=Table[0,{10}];
DeptDestinoF=Table[0,{10}];
Do[If[idorsal== 1,DeptOrigenF[[idorsal]]=TUMBES; DeptDestinoF[[idorsal]]=PIURA;];
If[idorsal== 2,DeptOrigenF[[idorsal]]=PIURA; DeptDestinoF[[idorsal]]=LAMBAYEQUE;];
If[idorsal== 3,DeptOrigenF[[idorsal]]=LAMBAYEQUE;DeptDestinoF[[idorsal]]=LALIBERTAD;];
If[idorsal== 4,DeptOrigenF[[idorsal]]=LALIBERTAD;DeptDestinoF[[idorsal]]=ANCASH;];
If[idorsal== 5,DeptOrigenF[[idorsal]]=ANCASH; DeptDestinoF[[idorsal]]=LIMA;];
If[idorsal== 6,DeptOrigenF[[idorsal]]=LIMA; DeptDestinoF[[idorsal]]=ICA;];
If[idorsal== 7,DeptOrigenF[[idorsal]]=ICA; DeptDestinoF[[idorsal]]=AREQUIPA;];
If[idorsal== 8,DeptOrigenF[[idorsal]]=AREQUIPA; DeptDestinoF[[idorsal]]=MOQUEGUA;];
If[idorsal== 9,DeptOrigenF[[idorsal]]=MOQUEGUA; DeptDestinoF[[idorsal]]=TACNA;];
If[idorsal==10,DeptOrigenF[[idorsal]]=PUNO; DeptDestinoF[[idorsal]]=AREQUIPA;];,{idorsal,10}];

Print["DeptOrigenF: Lista de Departamentos de Origen"];
Print[DeptOrigenF];
Print["DeptDestinoF: Lista de Departamentos de Destino"];
Print[DeptDestinoF];
(* Se especifican los departamentos (su número) que cuentan con transmisión de fibra. Básicamente son los departamentos ubicados en la costa. Se ha precisado cada uno de los tramos: Tumbes(17)-Piura(16), Piura(16)-Lambayeque(15),Lambayeque(15)-La Libertad(14), La Libertad(14)-Ancash(12), Ancash(12)-Lima(6), Lima(6)-Ica(4), Ica(4)-Arequipa(19), Arequipa(19)-Moquegua(22), Moquegua(22)-Tacna(24), Puno(23)-Arequipa(19). Además, en ese orden se precisa una lista de departamentos de origen {17,16,15,14,12,6,4,19,22,23} y de departamentos de destino {16,15,14,12,6,4,19,22,24,19} *)

Print["trafLDSF:Acumulación por Tramo de Fibra de los E1s de Voz LD"];
trafLDSF=Table[0,{10}];
Do[trafLDSF[[idorsal]]=traficoLDS[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoLDS[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];,{idorsal,10}];
Print[trafLDSF];
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s que pasan por el mismo. Ello implica sumar el total de segundos del punto a-b con el total de segundos del punto b-a contenido dentro de la variable traficoLDS estimada en la sección anterior (lista que 576 o 24*24 resultados) *)

Print["trafLDTotaf: Acumulación por Tramo del Tráfico Total en segundos "];
trafLDTotaf=trafLDSF;
Print[trafLDTotaf];
(* Para cada uno de los tramos de fibra se suma el total de E1s (de todos los tipos de comunicación identificados) que pasan por el mismo. *)

PRINT["ACUMULACIÓN DE E1s DE CIRCUITOS LD POR TRAMO DE FIBRA: RED DORSAL DE LA COSTA"]

Print["e1sLDTOTALF:Acumula por Tramo los E1s de todos los Servicios"];
e1sLDF=Table[0,{10}];
e1sLDadslF=Table[0,{10}];
e1sLDalqA1F=Table[0,{10}];
e1sLDalqB1F=Table[0,{10}];
e1sLDalqC1F=Table[0,{10}];
e1sLDalqABC2F=Table[0,{10}];
e1sLDalqABCnopF=Table[0,{10}];
Do[e1sLDF[[idorsal]]=trafLDTotaf[[idorsal]];
(* Estimación del total de E1s de voz que pasa por cada tramo de Fibra *)
e1sLDadslF[[idorsal]]=Round[traficoLDadsl[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoLDadsl[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de circuitos ADSL que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDadslF estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqA1F[[idorsal]]=Round[traficoAlqLDE1sA1[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoAlqLDE1sA1[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de circuitos de LD Rango A que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDalqA1F estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqB1F[[idorsal]]=Round[traficoAlqLDE1sB1[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoAlqLDE1sB1[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de circuitos de LD Rango B que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDalqB1F estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqC1F[[idorsal]]=Round[traficoAlqLDE1sC1[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoAlqLDE1sC1[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de circuitos de LD Rango C que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDalqC1 estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqABC2F[[idorsal]]=
Round[traficoAlqLDE1sABC2[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoAlqLDE1sABC2[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]];

```

```
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de otros circuitos de LD Rango A, B y C que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDalqABC2 estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqABCnopF[[idorsal]]=
Round[traficoAlqLDE1sABCnop[[nd*(DeptOrigenF[[idorsal]]-1)+DeptDestinoF[[idorsal]]]]+
traficoAlqLDE1sABCnop[[nd*(DeptDestinoF[[idorsal]]-1)+DeptOrigenF[[idorsal]]]]];,{idorsal,10}};
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de circuitos de LD no operadores que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la variable e1sLDalqABCnop estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDTOTALF=e1sLDF+e1sLDadslF+e1sLDalqA1F+e1sLDalqB1F+e1sLDalqC1F+e1sLDalqABC2F+e1sLDalqABCnopF;
(* Para cada uno de los tramos de fibra definidos, se suma el total de E1s de Todos los servicios *)
Print[MatrixForm[Transpose[{e1sLDF,e1sLDalqA1F,e1sLDalqB1F,e1sLDalqC1F,e1sLDalqABC2F,e1sLDalqABCnopF,{0,0,0,0,0,0,0,0,0}},e1sLDadslF,e1sLDTOTALF]],TableDirections->{Column,Row},TableAlignments->Right,TableHeadings->{{"TUMBES-PIURA","PIURA-LAMBAYEQUE","LAMBAYEQUE-LA LIBERTAD","LA LIBERTAD-ANCASH","ANCASH-LIMA","LIMA-ICA","ICA-AREQUIPA","AREQUIPA-MOQUEGUA","MOQUEGUA-TACNA","PUNO-AREQUIPA"},"Voz","Alq A","Alq B","Alq C","Alq ABC Otro","Alq No Op","Itx","ADSL","TOTAL"}]]];
```

PRINT["ESTIMACIÓN DEL NIVEL DE INVERSIÓN Y GASTO EN LA RED DORSAL DE FIBRA"]

```
Print["CargaFF: Carga Total en E1s por tramo de Fibra de Circuitos + Voz "];
CargaFF=e1sLDF+e1sLDalqA1F+e1sLDalqB1F+e1sLDalqC1F+e1sLDalqABC2F+e1sLDalqABCnopF+e1sLDadslF;
Print[CargaFF]
(* Especifica por tramo de Fibra el total de E1s asociados a los servicios interdepartamentales: Voz + Circuitos todos los servicios *)

Print["CargaTotalFF: Carga Total en E1s por tramo (Voz + Todo los Circuitos LD)"];
CargaTOTALFF=e1sLDF+e1sLDalqA1F+e1sLDalqB1F+e1sLDalqC1F+e1sLDalqABC2F+e1sLDalqABCnopF+e1sLDadslF;
Print[CargaTOTALFF]
(* Especifica por tramo de Fibra el total de E1s asociados a TODOS los servicios interdepartamentales: Voz + Circuitos todos los servicios*)

Print["numtributariosF"];
numtributariosF=Ceiling[CargaTOTALFF/21];
Print[numtributariosF];
(* Sobre la base del nivel de E1s totales (voz + circuitos) se estima el número de tributarios por tramo de fibra *)

Print["VelocidadFF"];
VelocidadFF=Table[0,{10}];
Do[VelocidadFF[[idorsal]]=0;
  If[CargaTOTALFF[[idorsal]]≠0,
Do[If[CargaTOTALFF[[idorsal]]>TxFibraLimites[[i]],VelocidadFF[[idorsal]]=i];,{i,Length[TxFibraLimites]}];
  If[VelocidadFF[[idorsal]]<Length[TxFibraLimites],
    VelocidadFF[[idorsal]]=VelocidadFF[[idorsal]]+1];];,{idorsal,10}};
Print[VelocidadFF];
(* Sobre la base del nivel de E1s totales (voz + circuitos) se estima el nivel de velocidad o capacidad que le corresponde a cada tramo de fibra *)

Print["CosteFijoLDFF"];
CosteFijoLDFF=Table[0,{10}];
Do[If[CargaTOTALFF[[idorsal]]≠0,
  CosteFijoLDFF[[idorsal]]=TxFibraCosteFijo[VelocidadFF[[idorsal]]];];
  If[VelocidadFF[[idorsal]]==Length[TxFibraLimites],
CosteFijoLDFF[[idorsal]]=CosteFijoLDFF[[idorsal]]*Ceiling[CargaTOTALFF[[idorsal]]/TxFibraLimites[[Length[TxFibraLimites]]];];,{idorsal,10}};
Print[CosteFijoLDFF];
(* Sobre la base de la velocidad o capacidad identificada en cada tramo se estima el costo unitario del equipo de Tx asociado a los mismos. Si la capacidad excede la capacidad máxima (en fibra STM64=63*64 E1s), se multiplica el costo del equipo de Tx por el número de STM64s que se necesitarían, el cual es equivalente al TOTAL de E1s entre el número total de E1s que hay en cada STM64 (63*64 E1s) *)

Print["CosteTributariosLDFF"];
CosteTributariosLDFF=Table[0,{10}];
Do[If[CargaTOTALFF[[idorsal]]≠0,
  CosteTributariosLDFF[[idorsal]]=TxFibraCosteTributario[VelocidadFF[[idorsal]]];];
  If[VelocidadFF[[idorsal]]==Length[TxFibraLimites],
    CosteTributariosLDFF[[idorsal]]=CosteTributariosLDFF[[idorsal]]*
Ceiling[CargaTOTALFF[[idorsal]]/TxFibraLimites[[Length[TxFibraLimites]]];];,{idorsal,10}};
Print[CosteTributariosLDFF];
(* Sobre la base de la velocidad o capacidad identificada en cada tramo se estima el costo del tributario asociado a los mismos. Si la capacidad excede la capacidad máxima (en fibra STM64=63*64 E1s), se multiplica el costo del equipo de Tx por el número de STM64s que se necesitarían, el cual es equivalente al TOTAL de E1s entre el número total de E1s que hay en cada STM64 (63*64 E1s) *)

Print["DistanciaLDFF"];
DistanciaLDFF=Table[0,{10}];
Do[DistanciaLDFF[[idorsal]]=distance[tandemd[[DeptOrigenF[[idorsal]]]],
  tandemd[[DeptDestinoF[[idorsal]]]]]*factnolinLD;,{idorsal,10}};
Print[DistanciaLDFF]
(* Distancia asociada a cada Tramo, equivale a la distancia entre las centrales Tandem de los respectivos departamentos *)

Print["invfiberLD1: Inversión Tottal por Equipos en Tramo de Fibra"];
invfiberLD1=(2*CosteFijoLDFF+2*CosteTributariosLDFF*numtributariosF+
  CosteFijoLDFF*Floor[DistanciaLDFF/hopfibra])*proteccionLD;
Print[invfiberLD1];
(* Equivale a la inversión asociada a los equipos dentro de cada tramo de fibra (equipo Tx, Tributarios y Repetidora). El nivel de inversión total se multiplica por el denominado factor de redudancia o protección *)
```



```

Print["invfiberLD2: Inversión Total por Cables en Tramos de Fibra"];
invfiberLD2=TxFibraCosteCableKmLD*DistanciaLDFF*compFOLD*proteccionLD;
Print[invfiberLD2];
(* Equivale a la inversión asociada a la cablería de cada tramo de fibra. El nivel de inversión total
equivale al producto de la distancia relevante y el costo del cable por Km. Dicho resultado se multiplica
por el factor de compartición con la red de la infraestructura con la red intradepartamental y el denominado
factor de protección *)

Print["invfiberLD3: Inversión Total por Obras Civiles en Tramos de Fibra"];
invfiberLD3=TxFibraCosteKmEnterrado*DistanciaLDFF*compFOLD*proteccionLD;
Print[invfiberLD3];
(* Equivale a la inversión asociada a las obras civiles de cada tramo de fibra. El nivel de inversión total
equivale al producto de la distancia relevante y el costo por Km. Dicho resultado se multiplica por el
factor de compartición con la red de la infraestructura con la red intradepartamental y el denominado factor
de protección *)

Print["costefiberLD1: Gasto Total por Equipos en Tramos de Fibra "];
costefiberLD1=(2*CosteFijoLDFF+2*CosteTributariosLDFF*numtributariosF+
CosteFijoLDFF*Floor[DistanciaLDFF/hopfibra])*FactorTx1*proteccionLD;
Print[costefiberLD1];
(* Equivale al gasto asociado a los equipos dentro de cada tramo de fibra (equipo Tx, Tributarios y
Repetidora). El nivel de inversión total se multiplica por el factor de anualización y el denominado factor
de redudancia o protección *)

Print["costefiberLD2: Gasto Total por Cables en Tramos de Fibra"];
costefiberLD2=TxFibraCosteCableKmLD*DistanciaLDFF*FactorTx2*compFOLD*proteccionLD;
Print[costefiberLD2];
(* Equivale al gasto asociado a la cablería de cada tramo de fibra. El nivel de inversión total equivale al
producto de la distancia relevante y el costo del cable por Km. Dicho resultado se multiplica por el factor
de anualización, el factor de compartición con la red de la infraestructura con la red intradepartamental y
el denominado factor de protección *)

Print["costefiberLD3: Gasto Total por Obras Civiles en Tramos de Fibra"];
costefiberLD3=TxFibraCosteKmEnterrado*DistanciaLDFF*FactorTx3*compFOLD*proteccionLD;
Print[costefiberLD3];
(* Equivale al gasto asociado a las obras civiles de cada tramo de fibra. El nivel de inversión total
equivale al producto de la distancia relevante y el costo por Km. Dicho resultado se multiplica por el
factor de anualización, el factor de compartición con la red de la infraestructura con la red
intradepartamental y el denominado factor de protección *)

PRINT["ACUMULACIÓN DE LOS GASTOS"];

Print["costeLDadsl: Gasto Total en Tramos de Fibra Imputable a ADSL"];
costeLDadsl=0;
Do[If[CargaTOTALFF[[idorsal]]#0,
  costeLDadsl=costeLDadsl+((costefiberLD1[[idorsal]]+costefiberLD2[[idorsal]]+
  costefiberLD3[[idorsal]])*(e1sLDadslF[[idorsal]]/CargaTOTALFF[[idorsal]]));],
,{idorsal,10}];
Print[costeLDadsl];
(* Equivale a la suma de los costos totales de todos los tramos (Tipo1+Tipo2+Tipo3) por el factor de
imputación del gasto al servicio de ADSL. Equivale entonces al gasto total a nivel de nacional por el tramo
de fibra imputable al servicio de ADSL *)

PRINT["ESTIMACIÓN DEL NIVEL DE GASTO EN LA RED DE RADIO RESPALDO DE LA RED DORSAL DE FIBRA"]

Print["VelocidadFR"];
VelocidadFR=Table[0,{10}];
Do[If[proteccionLD=1,
  If[CargaTOTALFF[[idorsal]]#0,
    Do[If[CargaTOTALFF[[idorsal]]>TxRadioLimites[[i]],
      VelocidadFR[[idorsal]]=i];,{i,Length[TxRadioLimites]}];
    If[VelocidadFR[[idorsal]]<Length[TxRadioLimites],
      VelocidadFR[[idorsal]]=VelocidadFR[[idorsal]]+1];];,{idorsal,10}];
Print[VelocidadFR];
(* Si se opta por protección vía radio: Sobre la base del nivel de E1s totales (voz + circuitos en la
modalidad elegida) se estima el nivel de velocidad o capacidad en rado que le corresponde a cada tramo de
radio a emplear como respaldo en la red de fibra *)

Print["CosteFijoLDFR"];
CosteFijoLDFR=Table[0,{10}];
Do[If[proteccionLD=1,
  If[CargaTOTALFF[[idorsal]]#0,
    CosteFijoLDFR[[idorsal]]=TxRadioCosteFijo[[VelocidadFR[[idorsal]]]]];];
  If[VelocidadFR[[idorsal]]==Length[TxRadioLimites],
    CosteFijoLDFR[[idorsal]]=CosteFijoLDFR[[idorsal]]*
    Ceiling[CargaTOTALFF[[idorsal]]/TxRadioLimites[[Length[TxRadioLimites]]]]];];,{idorsal,10}];
Print[CosteFijoLDFR];
(* Si se opta por protección vía radio: Sobre la base de la velocidad o capacidad identificada en cada tramo
se estima el costo unitario del equipo de Tx asociado a los mismos. Si la capacidad excede la capacidad
máxima (en radio STM1=63 E1s), se multiplica el costo del equipo de Tx por el número de STM1s que se
necesitarían, el cual es equivalente al TOTAL de E1s entre el número total de E1s que hay en cada STM1 (63
E1s) *)

Print["CosteTributariosLDFR"];
CosteTributariosLDFR=Table[0,{10}];
Do[If[proteccionLD=1,
  If[CargaTOTALFF[[idorsal]]#0,
    CosteTributariosLDFR[[idorsal]]=TxRadioCosteTributario[[VelocidadFR[[idorsal]]]]];];
  If[VelocidadFR[[idorsal]]==Length[TxRadioLimites],
    CosteTributariosLDFR[[idorsal]]=CosteTributariosLDFR[[idorsal]]*

```

```

Ceiling[CargaTOTALFF[[idorsal]]/TxRadioLimites[[Length[TxRadioLimites]]];];];,{idorsal,10}];
Print[CosteTributariosLDFR];
(* Si se opta por protección vía radio: Sobre la base de la capacidad identificada en cada tramo se estima
el costo del tributario asociado a los mismos. Si la capacidad excede la capacidad máxima (en radio STM1=63
E1s), se multiplica el costo del equipo de Tx por el número de STM1s que se necesitarían, el cual es
equivalente al TOTAL de E1s entre el número total de E1s que hay en cada STM1 (63 E1s) *)

Print["DistanciaLDFR"];
DistanciaLDFR=Table[0,{10}];
Do[If[proteccionLD==1,
      DistanciaLDFR[[idorsal]]=distance[tandemd[[DeptOrigenF[[idorsal]]]],
      tandemd[[DeptDestinoF[[idorsal]]]]*factnolinLD;];,{idorsal,10}];
Print[DistanciaLDFR];
(* Distancia asociada a cada Tramo, equivale a la distancia entre las centrales Tandem de los respectivos
departamentos *)

Print["invradioLD1F: Inversión Total por Respaldo en Radio en Tramos de Fibra"];
invradioLD1F=Table[0,{10}];
If[proteccionLD==1,
  invradioLD1F=(2*CosteFijoLDFR+2*CosteTributariosLDFR*numtributariosF+
  CosteFijoLDFR*Floor[DistanciaLDFR/hop])*0.6;];
Print[invradioLD1F];
(* Si se opta por protección vía radio: Equivale a la inversión asociada a los equipos dentro de cada tramo
de radio (equipo Tx, Tributarios y Repetidora). El nivel de inversión total se multiplica por el denominado
factor de redudancia o protección (60%) *)

Print["costeradioLD1F: Gasto Total por Respaldo en Radio en Tramos de Fibra"];
costeradioLD1F=Table[0,{10}];
If[proteccionLD==1,
  costeradioLD1F=(2*CosteFijoLDFR+2*CosteTributariosLDFR*numtributariosF+
  CosteFijoLDFR*Floor[DistanciaLDFR/hop])*FactorTx1*0.6;];
Print[costeradioLD1F];
(* Si se opta por protección vía radio: Equivale al gasto asociado a los equipos dentro de cada tramo de
radio (equipo Tx, Tributarios y Repetidora). El nivel de inversión total se multiplica por el factor de
anualización y el denominado factor de redudancia o protección (60%) *)

PRINT["ACUMULACIÓN DE LOS GASTOS"];

Print["costeLDadsl:Gasto Total Respaldo en Radio en Tramos de Fibra Imputable al Servicio de ADSL"];
Do[If[proteccionLD==1,
      If[CargaTOTALFF[[idorsal]]#0,
        costeLDadsl=costeLDadsl+(costeradioLD1F[[idorsal]]*
        (e1sLDadslF[[idorsal]]/CargaTOTALFF[[idorsal]]));];];
      ,{idorsal,10}];
Print[costeLDadsl];
(* Si se opta por protección vía radio: Equivale a la suma de los costos totales de todos los tramos (Tipo1)
por el factor de imputación del gasto al servicio de ADSL. Equivale entonces al gasto total a nivel de
nacional por el respaldo en radio para el tramo de fibra imputable al servicio de ADSL *)

PRINT["ACUMULACIÓN DE E1s DE VOZ POR TRAMO DE RADIO (MAYORMENTE EN LA SIERRA)"];

DeptOrigenR=Table[0,{11}];
DeptDestinoR=Table[0,{11}];
Do[If[iradio== 1,DeptOrigenR[[iradio]]=LALIBERTAD; DeptDestinoR[[iradio]]=CAJAMARCA;];
  If[iradio== 2,DeptOrigenR[[iradio]]=CAJAMARCA; DeptDestinoR[[iradio]]=AMAZONAS;];
  If[iradio== 3,DeptOrigenR[[iradio]]=LIMA; DeptDestinoR[[iradio]]=JUNIN;];
  If[iradio== 4,DeptOrigenR[[iradio]]=JUNIN; DeptDestinoR[[iradio]]=PASCO;];
  If[iradio== 5,DeptOrigenR[[iradio]]=PASCO; DeptDestinoR[[iradio]]=HUANUCO;];
  If[iradio== 6,DeptOrigenR[[iradio]]=HUANUCO; DeptDestinoR[[iradio]]=UCAYALI;];
  If[iradio== 7,DeptOrigenR[[iradio]]=JUNIN; DeptDestinoR[[iradio]]=HUANCAVELICA;];
  If[iradio== 8,DeptOrigenR[[iradio]]=HUANCAVELICA; DeptDestinoR[[iradio]]=AYACUCHO;];
  If[iradio== 9,DeptOrigenR[[iradio]]=AYACUCHO; DeptDestinoR[[iradio]]=APURIMAC;];
  If[iradio==10,DeptOrigenR[[iradio]]=PUNO; DeptDestinoR[[iradio]]=CUSCO;];
  If[iradio==11,DeptOrigenR[[iradio]]=APURIMAC; DeptDestinoR[[iradio]]=CUSCO;];,{iradio,11}];

Print["DeptOrigenR: Lista de Departamentos de Origen"];
Print[DeptOrigenR];
Print["DeptDestinoR: Lista de Departamentos de Destino"];
Print[DeptDestinoR];
(* Se especifican los departamentos (su número) que cuentan con transmisión de radio. Básicamente son los
departamentos ubicados en la sierra. Se ha precisado cada uno de los tramos: La Libertad(14)-Cajamarca(13),
Cajamarca(13)-Amazonas(11), Lima(6)-Junín(5), Junín(5)-Pasco(8), Pasco(8)-Huánuco(3), Huánuco(3)-
Ucayali(10), Junín(5)-Huancavelica(2), Huancavelica(2)-Ayacucho(1), Ayacucho(1)-Apurímac(18), Puno(23)-
Cusco(20), Apurímac(18)-Cusco(20). Además, en ese orden se presica una lista de departamentos de origen
{14,13,6,5,8,3,5,2,1,23,18} y de departamentos de destino {13,11,5,8,3,10,2,1,18,20,20} *)

Print["trafLDSR:Acumulación por Tramo de Radio de los E1s de Voz LD"];
trafLDSR=Table[0,{11}];
Do[trafLDSR[[iradio]]=traficoLDS[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]]+
  traficoLDS[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]];,{iradio,11}];
Print[trafLDSR];
(* Para cada uno de los tramos de radio definidos, se suma el total de segundos que pasan por el mismo. Ello
implica sumar el total de segundos del punto a-b con el total de segundos del punto b-a contenido dentro de
la variable traficoLDS estimada en la sección anterior (lista que 576 o 24*24 resultados) *)

Print["trafLDTotallR:Acumulación por Tramo de Radio del Tráfico Total en E1s"];
trafLDTotallR=trafLDSR;
Print[trafLDTotallR];
(* Para cada uno de los tramos de radio se suma el total de segundos (de todos los tipos de comunicación
identificados) que pasan por el mismo. *)

```

```
PRINT["ACUMULACIÓN DE E1s DE ALQUILER DE CIRCUITOS LD POR TRAMO DE RADIO: RED DE LA SIERRA"]
```

```
Print["Acumula por Tramo los E1s de todos los Servicios"];
e1sLDR=Table[0,{11}];
e1sLDalqA1R=Table[0,{11}];
e1sLDalqB1R=Table[0,{11}];
e1sLDalqC1R=Table[0,{11}];
e1sLDalqABC2R=Table[0,{11}];
e1sLDalqABCnopR=Table[0,{11}];
e1sLDadslR=Table[0,{11}];
Do[e1sLDR[[iradio]]=trafLDTOTALR[[iradio]];(*Ceiling[CalcTrunk[FactHC*trafLDTOTALR[[iradio]]/(60*60),prob]/30
/fiberfill];*)
(* Estimación del total de E1s de voz que pasa por cada tramo de Radio *)
e1sLDadslR[[iradio]]=Round[traficoLDadsl[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoLDadsl[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de ADSL que pasan por el mismo.
Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a contenido dentro de la
variable traficoAlqLDE1s estimada en la sección anterior (lista que 576 o 24*24 resultados) *)
e1sLDalqA1R[[iradio]]=Round[traficoAlqLDE1sA1[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoAlqLDE1sA1[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de circuitos de LD Rango A que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqA1R estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqB1R[[iradio]]=Round[traficoAlqLDE1sB1[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoAlqLDE1sB1[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de circuitos de LD Rango B que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqB1R estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqC1R[[iradio]]=Round[traficoAlqLDE1sC1[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoAlqLDE1sC1[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de circuitos de LD Rango C que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqC1R estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqABC2R[[iradio]]=Round[traficoAlqLDE1sABC2[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoAlqLDE1sABC2[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de otros circuitos de LD Rango A,
B y C que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto
b-a contenido dentro de la variable e1sLDalqABC2R estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqABCnopR[[iradio]]=Round[traficoAlqLDE1sABCnop[[nd*(DeptOrigenR[[iradio]]-1)+DeptDestinoR[[iradio]]]+
traficoAlqLDE1sABCnop[[nd*(DeptDestinoR[[iradio]]-1)+DeptOrigenR[[iradio]]]]];, {iradio, 11}];
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de circuitos de LD de No
Operadores que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del
punto b-a contenido dentro de la variable e1sLDalqABCnopR estimada en la sección anterior (lista que 576 o
24*24 resultados) *)
e1sLDTOTALR=e1sLDR+e1sLDadslR+e1sLDalqA1R+e1sLDalqB1R+e1sLDalqC1R+e1sLDalqABC2R+e1sLDalqABCnopR;
(* Para cada uno de los tramos de radio definidos, se suma el total de E1s de Todos los servicios *)
Print[MatrixForm[Transpose[{e1sLDR, e1sLDalqA1R, e1sLDalqB1R, e1sLDalqC1R, e1sLDalqABC2R, e1sLDalqABCnopR, {0, 0, 0, 0,
0, 0, 0, 0, 0, 0}, e1sLDadslR, e1sLDTOTALR}], TableDirections->{Column, Row}, TableAlignments->Right, TableHeadings->
{{"LA LIBERTAD-CAJAMARCA", "CAJAMARCA-AMAZONAS", "LIMA-JUNIN", "JUNIN-PASCO", "PASCO-HUANUCO", "HUANUCO-
UCAYALI",
"JUNIN-HUANCAVELICA", "HUANCAVELICA-AYACUCHO", "AYACUCHO-APURIMAC", "PUNO-CUSCO", "APURIMAC-CUSCO"}, {"Voz", "Alq
A", "Alq B", "Alq C", "Alq ABC Otro", "Alq No Op", "Itx", "ADSL", "TOTAL"}]]];
```

```
PRINT["ESTIMACIÓN DEL NIVEL DE INVERSIÓN Y GASTO EN LA RED DE RADIO DE LA SIERRA"]
```

```
Print["CargaR: Carga Total en E1s por tramo de Fibra para el todo servicio"];
CargaR=e1sLDR+e1sLDalqA1R+e1sLDalqB1R+e1sLDalqC1R+e1sLDalqABC2R+e1sLDalqABCnopR+e1sLDadslR;
Print[CargaR]
(* Especifica por tramo de Radio el totalde E1s asociados a los servicios interdepartamentales: Voz +
Circuitos todos los servicios *)

Print["CargaTOTALR: Carga Total en E1s por tramo (Voz + Todo los Circuitos LD)"];
CargaTOTALR=e1sLDR+e1sLDalqA1R+e1sLDalqB1R+e1sLDalqC1R+e1sLDalqABC2R+e1sLDalqABCnopR+e1sLDadslR;
Print[CargaTOTALR]
(* Especifica por tramo de Radio el total de E1s asociados a TODOS los servicios interdepartamentales: Voz +
Circuitos todos los servicios *)

Print["numtributariosR"]
numtributariosR=Ceiling[CargaTOTALR/21];
Print[numtributariosR]
(* Sobre la base del nivel de E1s totales (voz + circuitos ) se estima el número de tributarios por tramo de
Radio *)

Print["VelocidadR"]
VelocidadR=Table[0,{11}];
Do[If[CargaTOTALR[[iradio]]#0,
Do[If[CargaTOTALR[[iradio]]>TxRadioLimites[[i]], VelocidadR[[iradio]]=i];, {i, Length[TxRadioLimites]}];];
If[VelocidadR[[iradio]]<Length[TxRadioLimites],
VelocidadR[[iradio]]=VelocidadR[[iradio]]+1];, {iradio, 11}];
Print[VelocidadR]
(* Sobre la base del nivel de E1s totales (voz + circuitos) se estima el nivel de velocidad o capacidad que
le corresponde a cada tramo de Radio *)

Print["CosteFijoLDR"]
CosteFijoLDR=Table[0,{11}];
Do[If[CargaTOTALR[[iradio]]#0,
CosteFijoLDR[[iradio]]=TxRadioCosteFijo[[VelocidadR[[iradio]]]]];];
```



```

If[VelocidadR[[iradio]]==Length[TxRadioLimites],

CosteFijoLDR[[iradio]]=CosteFijoLDR[[iradio]]*Ceiling[CargaTOTALR[[iradio]]/TxRadioLimites[[Length[TxRadioLim
ites]]];];,{iradio,11}};
Print[CosteFijoLDR]
(* Sobre la base de la velocidad o capacidad identificada en cada tramo se estima el costo unitario del
equipo de Tx asociado a los mismos. Si la capacidad excede la capacidad máxima (en radio STM1=63 E1s), se
multiplica el costo del equipo de Tx por el número de STM1s que se necesitarían, el cual es equivalente al
TOTAL de E1s entre el número total de E1s que hay en cada STM1 (63 E1s) *)

Print["CosteTributariosLDR"]
CosteTributariosLDR=Table[0,{11}];
Do[If[CargaTOTALR[[iradio]]#0,
    CosteTributariosLDR[[iradio]]=TxRadioCosteTributario[[VelocidadR[[iradio]]]];];
    If[VelocidadR[[iradio]]==Length[TxRadioLimites],
        CosteTributariosLDR[[iradio]]=CosteTributariosLDR[[iradio]]*Ceiling[CargaTOTALR[[iradio]]/
        TxRadioLimites[[Length[TxRadioLimites]]];];,{iradio,11}};
Print[CosteTributariosLDR]
(* Sobre la base de la velocidad o capacidad identificada en cada tramo de Radio se estima el costo del
tributario asociado a los mismos. Si la capacidad excede la capacidad máxima (en radio STM1=63 E1s), se
multiplica el costo del equipo de Tx por el número de STM1s que se necesitarían, el cual es equivalente al
TOTAL de E1s entre el número total de E1s que hay en cada STM1 (63 E1s) *)

Print["DistanciaLDR"]
DistanciaLDR=Table[0,{11}];
Do[DistanciaLDR[[iradio]]=distance[tandemd[[DeptOrigenR[[iradio]]]],
tandemd[[DeptDestinoR[[iradio]]]]*factnolinLD;,{iradio,11}};
Print[DistanciaLDR]
(* Distancia asociada a cada Tramo, equivale a la distancia entre las centrales Tandem de los respectivos
departamentos *)

Print["invradioLD1R: Inversión Total en la Red de Radio en la Sierra"];
invradioLD1R=Table[0,{11}];
invradioLD1R=(2*CosteFijoLDR+2*CosteTributariosLDR*numtributariosR+CosteFijoLDR*Floor[DistanciaLDR/hop])*1.6;
Print[invradioLD1R];
(* Equivale a la inversión asociada a los equipos dentro de cada tramo de radio (equipo Tx, Tributarios y
Repetidora). El nivel de inversión total se multiplica por el denominado factor de redudancia o protección
(160%) *)

Print["costeradioLD1R: Gasto Total en la Red de Radio en la Sierra "]
costeradioLD1R=(2*CosteFijoLDR+2*CosteTributariosLDR*numtributariosR+
CosteFijoLDR*Floor[DistanciaLDR/hop])*FactorTx*1.6;
Print[costeradioLD1R]
(* Equivale al gasto asociado a los equipos dentro de cada tramo de radio (equipo Tx, Tributarios y
Repetidora). El nivel de inversión total se multiplica por el factor de anualización y el denominado factor
de redudancia o protección (160%) *)

PRINT["ACUMULACIÓN DE LOS GASTOS"];

Print["costeLDadsl: Gasto Total Acumulado en la red Intradepartamental (Fibra+Radio) ADSL"];
Do[If[CargaTOTALR[[iradio]]#
0,costeLDadsl=costeLDadsl+(costeradioLD1R[[iradio]]*(e1sLDadslR[[iradio]]/CargaTOTALR[[iradio]]));];,{iradio,
11}};
Print[costeLDadsl]
(* Equivale a la suma de los costos totales de todos los tramos (Tipo1) por el factor de imputación del
gasto al servicio de ADSL. Equivale entonces al gasto total a nivel de nacional por el tramo de radio
imputable al servicio de ADSL. Nótese que dicho resultado se acumula con el valor de costeLDadsl estimada
antes en Fibra *)

PRINT["ACUMULACIÓN DE E1s DE VOZ POR TRAMOS DE TX VÍA SATELITE (MAYORMENTE EN LA SELVA)"]

DeptOrigenS=Table[0,{3}];
DeptDestinoS=Table[0,{3}];
Do[If[isat==1,DeptOrigenS[[isat]]=LIMA;DeptDestinoS[[isat]]=LORETO;];
    If[isat==2,DeptOrigenS[[isat]]=LIMA;DeptDestinoS[[isat]]=SANMARTIN;];
    If[isat==3,DeptOrigenS[[isat]]=LIMA;DeptDestinoS[[isat]]=MADREDEDIOS;];,{isat,3}};
Print["DeptOrigenS: Lista de Departamentos de Origen"];
Print[DeptOrigenS];
Print["DeptDestinoS: Lista de Departamentos de Destino"];
Print[DeptDestinoS];
(* Se especifican los departamentos (su número) que cuentan con transmisión vía satélite. Básicamente son
los departamentos ubicados en la selva. Se ha precisado cada uno de los tramos: Lima(6)-Loreto(7), Lima(6)-
San Martín(9), Lima(18)-Madre de Dios(21). Además, en ese orden se presica una lista de departamentos de
origen {6,6,6} y de departamentos de destino {7,9,21} *)

Print["trafLDSS:Acumulación por Tramo Satélital de los E1s de Voz LD"]
trafLDSS=Table[0,{3}];
Do[trafLDSS[[isat]]=traficoLDS[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+
traficoLDS[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]];,{isat,3}};
Print[trafLDSS]
(* Para cada uno de los tramos de satélite definidos, se suma el total de segundos que pasan por el mismo.
Ello implica sumar el total de segundos del punto a-b con el total de segundos del punto b-a contenido
dentro de la variable traficoLDS estimada en la sección anterior (lista que 576 o 24*24 resultados) *)

Print["trafLDTotals: Acumulación por Tramo Satelital del Tráfico Total en segundos "]
trafLDTotals=trafLDSS;
Print[trafLDTotals]
(* Para cada uno de los tramos vía satélite se suma el total de segundos (de todos los tipos de comunicación
identificados) que pasan por el mismo *)

PRINT["ACUMULACIÓN DE E1s DE CIRCUITOS LD POR TRAMO SATELITAL: RED DE LA SELVA"]

```

```
Print["Acumulación por Tramo de E1s"];
e1sLDS=Table[0,{3}];
e1sLDadslS=Table[0,{3}];
e1sLDalqA1S=Table[0,{3}];
e1sLDalqB1S=Table[0,{3}];
e1sLDalqC1S=Table[0,{3}];
e1sLDalqABC2S=Table[0,{3}];
e1sLDalqABCnopS=Table[0,{3}];
Do[e1sLDS[[isat]]=trafLDTotals[[isat]];
(* Estimación del total de E1s de voz que pasa por cada tramo de Satélite *)
e1sLDadslS[[isat]]=Round[traficoLDadsl[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+traficoLDadsl[[nd*(
DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de circuitos LD ADSL que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDadslS estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqA1S[[isat]]=Round[traficoAlqLDE1sA1[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+traficoAlqLDE1
sA1[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de circuitos LD Rango A que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqA1S estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqB1S[[isat]]=Round[traficoAlqLDE1sB1[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+traficoAlqLDE1
sB1[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de circuitos LD Rango B que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqB1S estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqC1S[[isat]]=Round[traficoAlqLDE1sC1[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+traficoAlqLDE1
sC1[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de circuitos LD Rango C que
pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del punto b-a
contenido dentro de la variable e1sLDalqC1S estimada en la sección anterior (lista que 576 o 24*24
resultados) *)
e1sLDalqABC2S[[isat]]=Round[traficoAlqLDE1sABC2[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+traficoAlq
LDE1sABC2[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de otros circuitos LD Rango
A, B y C que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del
punto b-a contenido dentro de la variable e1sLDalqABC2S estimada en la sección anterior (lista que 576 o
24*24 resultados) *)
e1sLDalqABCnopS[[isat]]=Round[traficoAlqLDE1sABCnop[[nd*(DeptOrigenS[[isat]]-1)+DeptDestinoS[[isat]]]]+trafic
oAlqLDE1sABCnop[[nd*(DeptDestinoS[[isat]]-1)+DeptOrigenS[[isat]]]]];
(* Para cada uno de los tramos vía satélite definidos, se suma el total de E1s de circuitos LD de No
Operadores que pasan por el mismo. Ello implica sumar el total de E1s del punto a-b con el total de E1s del
punto b-a contenido dentro de la variable e1sLDalqABCnopS estimada en la sección anterior (lista que 576 o
24*24 resultados) *)
,{isat,3}];
e1sLDTOTALS=e1sLDS+e1sLDadslS+e1sLDalqA1S+e1sLDalqB1S+e1sLDalqC1S+e1sLDalqABC2S+e1sLDalqABCnopS;
(* Para cada uno de los tramos satelitales definidos, se suma el total de E1s de Todos los servicios *)
Print[MatrixForm[Transpose[{e1sLDS,e1sLDalqA1S,e1sLDalqB1S,e1sLDalqC1S,e1sLDalqABC2S,e1sLDalqABCnopS},{0,0,0},
e1sLDadslS,e1sLDTOTALS]],TableDirections->{Column,Row},TableAlignments->Right,TableHeadings->{{"LIMA-
LORETO","LIMA-SAN MARTIN","LIMA-MADRE DE DIOS"},{"Voz","Alq A","Alq B","Alq C","Alq ABC Otro","Alq No
Op","Itx","ADSL","TOTAL"}}]]];
```

PRINT["ESTIMACIÓN DEL NIVEL DE INVERSIÓN Y GASTO EN LA RED SATELITAL DE LA SELVA"]

```
Print["CargaTotals: Carga Total en E1s por tramo (Voz + Todos Servicios LD)"];
CargaTOTALS=e1sLDS+e1sLDalqA1S+e1sLDalqB1S+e1sLDalqC1S+e1sLDalqABC2S+e1sLDalqABCnopS+e1sLDadslS;
Print[CargaTOTALS]
(* Especifica por tramo satelital el total de E1s asociados a TODOS los servicios interdepartamentales: Voz
+ Circuitos todos los servicios *)

Print["invsatLD1: Equipos y Antenas"]
invsatLD1=2*(satantena+sateqtx);
Print[invsatLD1]
(* Equivale a la inversión asociada a los equipos dentro de cada tramo vía satélite (equipo Tx y Antena) *)

Print["costesatLD1: Equipos, Antenas y Espectro satelital"]
costesatLD1=2*(satantena+sateqtx)*FactorTx1+2*Ceiling[CargaTOTALS*compresDCME]*numMHZporMb*costesatporMHz;
Print[costesatLD1]
(* Equivale al gasto asociado a los equipos dentro de cada tramo vía satélite (equipo Tx y Antena) por el
factor de imputación del gasto al servicio elegido. El nivel de inversión total se multiplica por el factor
de anualización *)
```

PRINT["ACUMULACIÓN DE LOS GASTOS"];

```
Print["costeLDadsl: Gasto Total Acumulado en la red Intradepartamental (Fibra+Radio+Satelite) ADSL"]
Do[If[CargaTOTALS[[isat]]!=0,
costeLDadsl=costeLDadsl+(costesatLD1[[isat]]*(e1sLDadslS[[isat]]/CargaTOTALS[[isat]]));,,{isat,3}];
Print[costeLDadsl]
(* Equivale a la suma de los costos totales de todos los tramos satelitales imputable al servicio de ADSL.
Nótese que dicho resultado se acumula con el valor de costeLDadsl estimada antes en Fibra y Radio *)
```

CÁLCULOS FINALES


```

Print["InversionDSLAM"];
InversionDSLAM=CostoDSLAMConTarjetaNT*EvPrecios[[1,2]];
Print[InversionDSLAM];

Print["InversionATM"];
InversionATM=CostoTotalPopATM*EvPrecios[[2,2]];
Print[InversionATM];

Print["InversionEnergiaDSLAM"];
InversionEnergiaDSLAM=CostoEnergiaDSLAM;
Print[InversionEnergiaDSLAM];

Print["InversionEnergiaATM"];
InversionEnergiaATM=CostoEnergiaATM;
Print[InversionEnergiaATM];

```

PRINT["CALCULO DEL PORCENTAJE DE ASIGNACION A PUERTOS"];

```

Print["CTDSLAM"];
InvDSLAM = {Plus @@ InversionDSLAM, Plus @@ InversionEnergiaDSLAM, Plus @@ InversionDSLAM * CosteActivs[[1, 2]]};
FactAnualDSLAM = {FactAnualizCxDSLAM, FactAnualizPtaSecENERGIA, FactAnualizPtaSecINMOB};
CTDSLAM = FullSimplify[InvDSLAM.FactAnualDSLAM];
Print[CTDSLAM];
(*Inversión en DSLAM, Anualizando*)

Print["CTATM"];
InvATM = {Plus @@ InversionATM, Plus @@ InversionEnergiaATM, Plus @@ InversionATM * CosteActivs[[1, 2]]};
FactAnualATM = {FactAnualizCxNodoATM, FactAnualizPtaSecENERGIA, FactAnualizPtaSecINMOB};
CTATM = FullSimplify[InvATM.FactAnualATM];
Print[CTATM];
(*Inversión en ATM, Anualizando*)

Print["CTTxadsl"];
Print[CTTxadsl];
(*Gasto Anualizado en Transmisión Intradepartamental, atribuible a Puertos*)

Print["costeLDadsl"];
Print[costeLDadsl];
(*Gasto Anualizado en Transmisión Interdepartamental, atribuible a Puertos*)

Print["Repartición"];
CTRepartATM = Round[CTATM / (CTDSLAM + CTATM + CTTxadsl + costeLDadsl) * 100, 0.1];
Print["ATM: ", CTRepartATM, "%"];
CTRepartTxLoc = Round[CTTxadsl / (CTDSLAM + CTATM + CTTxadsl + costeLDadsl) * 100, 0.1];
Print["TxLoc: ", CTRepartTxLoc, "%"];
CTRepartTxLD = Round[costeLDadsl / (CTDSLAM + CTATM + CTTxadsl + costeLDadsl) * 100, 0.1];
Print["TxLD: ", CTRepartTxLD, "%"];
(*Se calcula cada elemento por separado *)
CTRepart = CTRepartTxLD + CTRepartTxLoc + CTRepartATM;
Print["Total: ", CTRepart, "%"];

```