

NACIONES UNIDAS

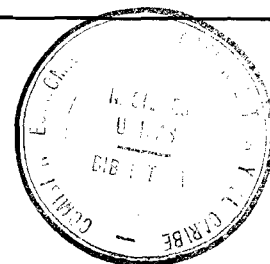
COMISION ECONOMICA
PARA AMERICA LATINA
Y EL CARIBE - CEPAL



Distr.
LIMITADA

LC/MEX/L.219
19 de marzo de 1993

ORIGINAL: ESPAÑOL



SIMULADOR INTERACTIVO DE FLUJOS DE POTENCIA

(Manual del programador)

23 JAN 1994

Este documento fue preparado por el señor José Horacio Tovar Hernández, consultor de la Unidad de Energía de la CEPAL. No fue sometido a revisión editorial.



INDICE

	<u>Página</u>
PRESENTACION	1
MANUAL DE PROGRAMACION DEL SIMULADOR DE FLUJOS DE POTENCIA ...	3
1. Introducción	3
2. Organización del Simulador	3
3. Descripción del Programa Principal	5
4. Descripción de Subprogramas para Actividades de Entrada	11
a) Subprograma PORTAD	11
b) Subprograma LEDATO	20
c) Subprograma PANTAL	35
5. Descripción de Subprogramas para Actividades de Salida	39
a) Subprograma FINALE	39
b) Subprograma DATREP	60
c) Subprograma RESREP	74
d) Subprograma TOPCAM	95
6. Subprogramas de Enlace	136
a) Subprograma CLEAR	136
b) Subprograma ENCA	137
c) Subprograma MENU	138
d) Subprograma CUADRO	150
e) Subprograma INIMEN	151
f) Subprograma VERIF	153
g) Subprograma SINO	156
h) Subprograma SALIAS	157
i) Subprograma CICLO	159
7. Subprogramas Algorítmicos	160
a) Subprograma GRAFIC	160
b) Subprograma INICFC	164

c)	Subprograma CIRCUI	165
d)	Subprograma YNODAL	169
e)	Subprograma ASIGNA	175
f)	Subprograma FORMPR	180
g)	Subprograma INICIA	181
h)	Subprograma MODEBP	185
i)	Subprograma REINIC	189
j)	Subprograma SOLFLU	193
k)	Subprograma ORDENA	206
l)	Subprograma DERPAP	212
m)	Subprograma FACTBP	216
n)	Subprograma PNODAL	219
o)	Subprograma DESVIA	222
p)	Subprograma SOLBBP	226
q)	Subprograma CORRIG	229
r)	Subprograma REVNPV	231
8.	Referencias	239

PRESENTACION

Como una continuación de la cooperación técnica que la Comisión Económica para América Latina y el Caribe (CEPAL) proporciona a las empresas eléctricas nacionales de América Central, contrató los servicios del consultor José Horacio Tovar Hernández para desarrollar un simulador interactivo de flujos de potencia. En este documento se proporciona el manual del programador, que será de utilidad a los profesionales interesados en incorporarle mejoras o adecuaciones. En un volumen por separado 1/ se presentan los manuales del usuario y técnico del simulador.

Anteriormente, la CEPAL ya había distribuido un modelo digital sobre el mismo tema; 2/ sin embargo, sobre la base de la experiencia en el uso del modelo anterior y de los comentarios que se han recibido de los técnicos de las empresas eléctricas del Istmo Centroamericano, se optó por desarrollar este nuevo simulador que, al igual que el anterior, se puede correr en una microcomputadora con requerimientos mínimos: sistema operativo DOS en cualquier versión posterior a la 2.0, y 640 kbytes de memoria RAM. Debido a que se basa en el método desacoplado rápido de Newton y, una vez instalado, se activa con DR.

Las características más importantes de este simulador son las siguientes: a) es muy "amigable"; b) ofrece gran flexibilidad de reportes de datos y resultados; c) tiene una subrutina de cambios que posibilita efectuar múltiples corridas, partiendo de un caso base, ya que se pueden modificar, remover y agregar datos y parámetros de la corrida, y d) incluye reportes específicos sobre pérdidas, que podrían ser útiles para estudios técnicos y económicos en el área de planificación, operación y distribución.

Al igual que para el modelo anterior, con el propósito de que estos desarrollos se conviertan también en una transferencia tecnológica, además de la versión ejecutable, se transferirá a las empresas eléctricas el código fuente del simulador.

1/ Véase, CEPAL, *Simulador interactivo de flujos de potencia (Manuales del usuario y técnico)* (LC/MEX/L.217), 1 de febrero de 1993.

2/ Véanse, CEPAL, *Programa de flujos de potencia. Volumen I. Manuales del usuario y técnico, y Volumen II. Manual del programa y código fuente* (LC/MEX/R.198), 25 de enero de 1990.

Como es usual en este tipo de herramientas, mediante su aplicación surgen nuevos requerimientos de mejoras y ampliaciones, por lo que en una fase posterior podrían incluirse las sugerencias y comentarios que se reciban de los técnicos que las utilicen en el desempeño de sus funciones. Se agradecerá que las dirijan a la Subsección de la CEPAL en México.

MANUAL DE PROGRAMACION DEL SIMULADOR DE FLUJOS DE POTENCIA

1. Introducción

En este manual se describen los subprogramas incluidos en el simulador de flujos de potencia. Primeramente, se presenta la organización del simulador considerando el uso de subprogramas por algún otro. Posteriormente, se describe cada uno de ellos, incluyendo la función de cada variable de entrada y/o salida, así como los procesos realizados en los mismos, en base a su código fuente.

2. Organización del Simulador

El simulador de flujos de potencia hace uso de un programa principal llamado DR.FOR, y un conjunto de 34 subprogramas, donde cada uno de ellos puede utilizar a otros. Desde este punto de vista, y partiendo del programa principal, se genera el diagrama mostrado en la Figura 1.

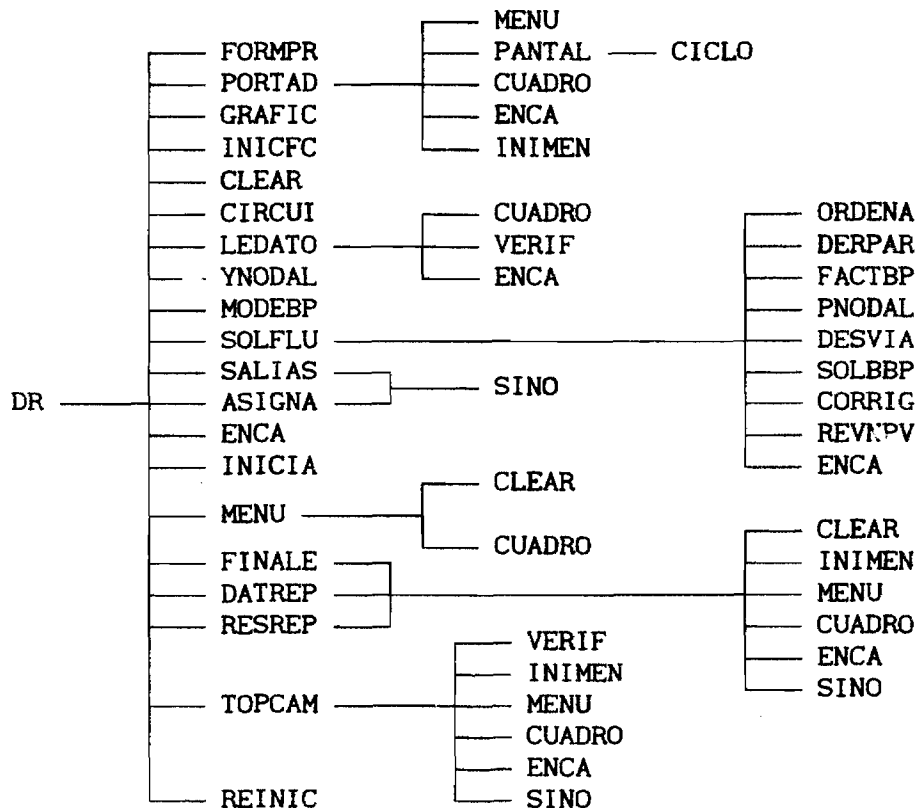


Figura 1. Organización de subprogramas que integran al simulador de flujos de potencia.

Las actividades del simulador se han clasificado en cuatro conjuntos:

1. De entrada.
2. De salida.
3. De enlace.
4. Algorítmicas.

Los subprogramas involucrados en cada actividad se muestran en la Tabla 1.

Tabla 1. Clasificación de los subprogramas que componen al simulador de flujos DR, de acuerdo al tipo de actividad que realizan.

ACTIVIDADES				
SUBPROGRAMA	ENTRADA	SALIDA	ENLACE	ALGORITMICA
GRAFIC				<input type="checkbox"/>
PORTAD	<input type="checkbox"/>			
INICFC				<input type="checkbox"/>
CLEAR			<input type="checkbox"/>	
LEDATO	<input type="checkbox"/>			
CIRCUI				<input type="checkbox"/>
YNODAL				<input type="checkbox"/>
FORMPR				<input type="checkbox"/>
ASIGNA				<input type="checkbox"/>
INICIA				<input type="checkbox"/>
MODEBP				<input type="checkbox"/>
SOLFLU				<input type="checkbox"/>
ENCA			<input type="checkbox"/>	
MENU			<input type="checkbox"/>	
FINALE		<input type="checkbox"/>		
DATREP		<input type="checkbox"/>		
RESREP		<input type="checkbox"/>		
TOPCAM		<input type="checkbox"/>		
REINIC				<input type="checkbox"/>
CUADRO			<input type="checkbox"/>	
INIMEN			<input type="checkbox"/>	
VERIF			<input type="checkbox"/>	
SINO			<input type="checkbox"/>	
ORDENA				<input type="checkbox"/>
DERPAR				<input type="checkbox"/>
FACTBP				<input type="checkbox"/>
PNODAL				<input type="checkbox"/>
DESVIA				<input type="checkbox"/>
SOLBBP				<input type="checkbox"/>
CORRIC				<input type="checkbox"/>
REVPV				<input type="checkbox"/>
SALIAS			<input type="checkbox"/>	
PANTAL	<input type="checkbox"/>			
CICLO			<input type="checkbox"/>	

Debe hacerse notar que este simulador de flujos está constituido por algunos subprogramas desarrollados por el Dr. Ricardo Mota Palomino [1], ya sea con modificaciones y/o adiciones. Esto mismo ocurre con la organización de los distintos reportes de flujos también cuya estructura ha sido desarrollada siguiendo su mismas ideas. Tales subprogramas, en sus primeras líneas de comentarios contiene el nombre del autor.

3. Descripción del Programa Principal

El programa principal se llama DR, y su propósito es coordinar todas las actividades del simulador. Este programa, en forma de bloques funcionales, tiene la estructura mostrada en la Figura 2.

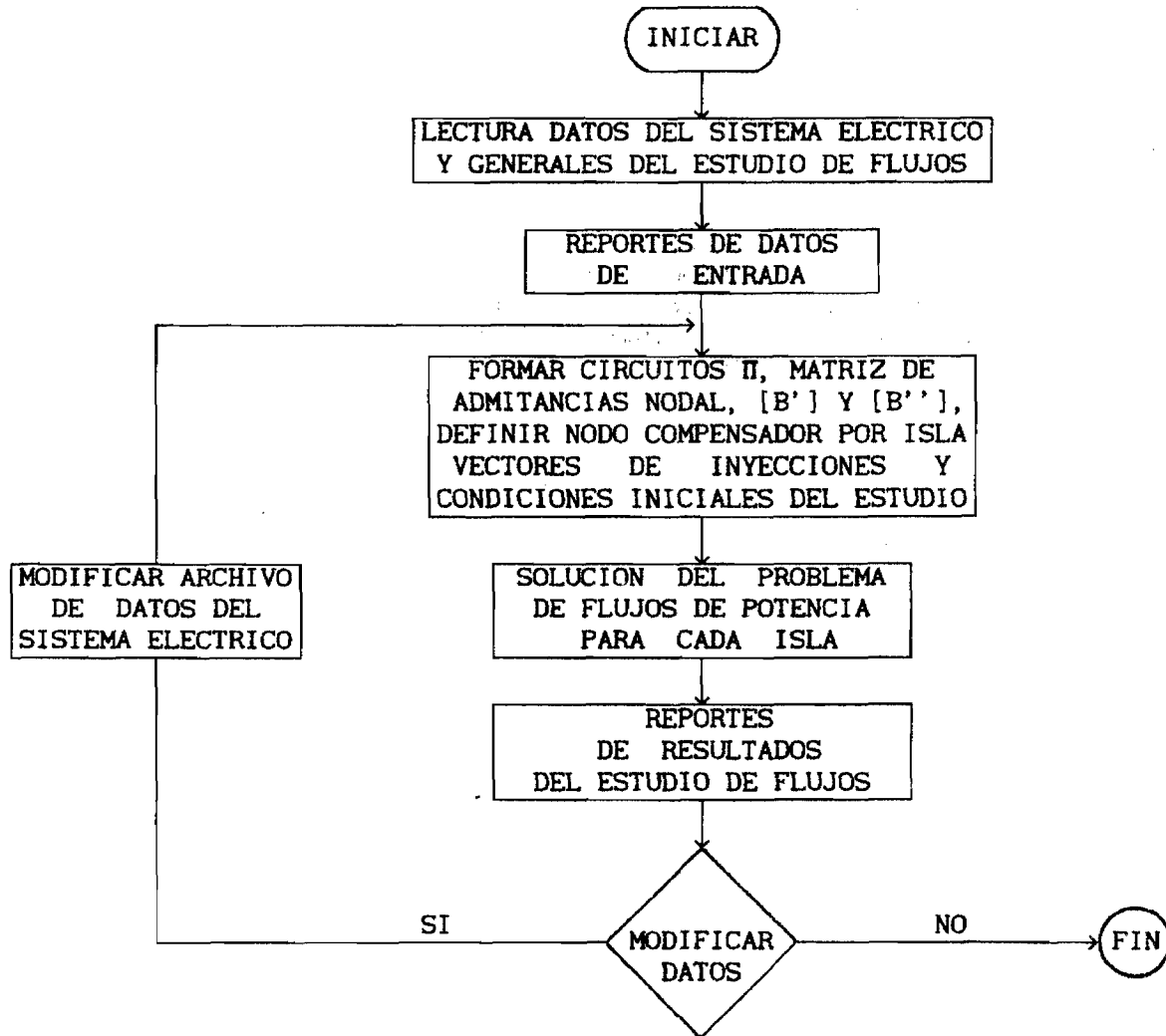


Figura 2. Estructura funcional del programa principal DR.

A continuación, se muestra el código del programa principal DR, describiendo en cuadros encerrados, cuyo contenido no es parte del mismo, cada uno de los procesos que lo constituyen.

```
C*****
C   PROGRAMA PRINCIPAL DEL METODO DESACOPLADO RAPIDO.
C   NOMBRE DEL PROGRAMA: DR
C   ELABORADO POR: JOSE HORACIO TOVAR HERNANDEZ.
C   OCTUBRE DE 1992.
C*****
```

Definición y dimensionamiento de variables enteras.

```
INTEGER NE(1000), NR(1000), NETR(300), NRTR(300), NLT(550), TIPO(550),
*   INIC(550), ICOL(2550), JICOL(2550), ISP(2550), TIPGEN(150),
*   ISLACK(10), INICPR(550), IFINPR(2000), JINIC(550), NAU(150),
*   IPOSPR(2000), INISLA(10), ISLA(550), NPV(150), NSHUNT(150),
*   JISP(2550), NEXT(2550), IFACT(550), NCEVV(150), NORDEN(550),
*   NODFUE(15), OPCION, SALIDA, ABORTA, DEPEND(15), ENTRA, PRINTR
```

Definición y dimensionamiento de variables reales.

```
REAL R(1000), X(1000), YPQ2E(1000), YPQ2R(1000), PG(550), QG(550),
*   R1(1000), X1(1000), BPRIMA(2550), BBPRIM(2550), GPQ2E(1000),
*   PC(550), QC(550), SHUNT(150), VIM(550), TAP(300), VESP(150),
*   GPQ2R(1000), YBUSR(2550), YBUSI(2550), VRE(550), QGMIN(150),
*   QGMAX(150), PCAL(550), QCAL(550), YM(150), VOLT(550),
*   ANGULO(550), FPCISLA(10), FQCISLA(10), PSLACK(10)
REAL*8 VALNUM(15)
```

Definición y dimensionamiento de variables tipo alfanumérico (carácter)

```
CHARACTER LISTOP(15)*50, ENCABE*50, ALTERN*50, DNOMB(550)*8
CHARACTER TIPVAL(15)*3, VALCHA(15)*12, ARCH*40, ARCH2*40, ARCH3*40
CHARACTER ESC*1, INVERS*3, LETR1*70, LETR2*70, NOMB1*8, NOMB2*8
```

Definición de los bloques comunes MENU12, TRANSF y CIRCUI.

```
COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
COMMON /CIRCUI/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
```

C

Se define salida por error en apertura de archivo de datos del sistema eléctrico (SALIDA = 1 para salir del simulador).
El valor de ABORTA define salida por error en algún proceso (ABORTA=1).
Se define el valor de la variable alfanumerica ESC para crear efectos de pantalla mediante secuencias de escape.

SALIDA = 0

```
ABORTA = 0
ESC = CHAR(27)
```

Se llama al subprograma PORTAD, donde se definen nombres de archivos de lectura de datos, de escritura de datos de entrada, así como los dispositivos correspondientes.

```
CALL PORTAD (ARCH, ARCH2, ARCH3, LISTIN, PRINTR, ABORTA, SALIDA)
IF (SALIDA.EQ.1) GO TO 1000
IF (ABORTA.EQ.0) GO TO 5
```

Se llama al subprograma SALIAS, donde se confirma la salida definitiva del simulador. Si INDSAL = 1, se continúa con la simulación.

```
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 5
GO TO 888
```

Se llama al subprograma LEDATO, donde se efectúa la lectura de datos - del sistema eléctrico de potencia.

```
CALL LEDATO (ARCH, DNOMB, LETR1, LETR2, NE, NR, NPV, NAU, NSHUNT, R, X,
*          YPQ2E, PC, QC, PG, VESP, QGMAX, QGMIN, SHUNT, QG, NNODOS,
*          NLLINES, NNGEN, NSHNTS, NCEVS, LISTIN, PRINTR, TIPGEN,
*          ARCH2, ARCH3, IERR, TIPO, TOLP, TOLQ, NMAXIT, SBASE, TOLREV)
```

Si existe un error en la lectura de datos, la simulación se termina en forma inmediata, esto ocurre cuando IERR = 1.

```
IF (IERR.EQ.1) GO TO 1000
```

Se copian variables para usos posteriores.

```
LINES = NLLINES
NODOS = NNODOS
```

Se define el valor de KFUERA, contador de nodos que se desconectan al - modificar datos del sistema eléctrico en el subprograma TOPCAM.

```
KFUERA = 0
```

Se inicializan valores de ramas en derivación de circuitos π .

```
DO I = 1, NLLINES
  YPQ2R(I) = YPQ2E(I)
  GPQ2E(I) = 0.0
  GPQ2R(I) = 0.0
END DO
```

Se llama al subprograma FORMPR para definir listas encadenadas por - nodo con respecto a circuitos π .

```
10 CALL FORMPR (NNODOS)
```

Se llama al subprograma GRAFIC que define el número de islas eléctricas

```
CALL GRAFIC (NNODOS, NISLAS, INISLA, ISLA, NODFUE, KFUERA)
```

Se llama al subprograma INICFC para inicializar factores de modificación de carga por isla.

```
CALL INICFC (NISLAS, FPCISLA, FQCISLA)
```

C

En ASIGNA se define el nodo compensador por isla eléctrica.

```
CALL ASIGNA (NISLAS, ISLA, INISLA, PG, TIPO, ISLACK, NNODOS, NNGEN,
*          DNOMB, NPV, PSLACK)
```

C

Se llama al subprograma CIRCUI para construir circuitos π .

```
CALL CIRCUI (YPQ2E, YPQ2R, R, X, R1, X1, GPQ2E, GPQ2R)
```

C

En YNODAL se construye la matriz de admitancias nodal del sistema.

```
CALL YNODAL (R1, X1, YPQ2E, YPQ2R, SHUNT, NNODOS, NSHNTS, NSHUNT, INIC,
*          ICOL, YBUSR, YBUSI, ISP, NLT, IERR, GPQ2E, GPQ2R, LONG,
*          KFUERA, NODFUE)
```

C

Si existe algún error al formar la matriz de admitancias nodal, se termina la sesión de simulación.

```
IF(IERR.EQ.1) GO TO 1000
```

C

En MODEBP se construyen los modelos generales del método desacoplado rápido, considerando que todos los nodos son tipo PQ.

```
CALL MODEBP (X, BPRIMA, BBPRIM, NNODOS, INIC, ICOL, ISP, YBUSI, LONG)
```

C

Se inicializan valores de voltaje y ángulo y se definen vectores de inyección.

```
CALL INICIA (VRE, VIM, VOLT, ANGULO, PG, PC, QG, QC, NNODOS, QGMIN,
*          QGMAX, NNGEN, VESP, NPV, NISLAS, ISLACK, NISLAS, ISLA)
```

C

C SE RESUELVE EL PROBLEMA DE FLUJOS PARA CADA ISLA.

C

```
DO 30 KISLA = 1, NISLAS
```

SOLFLU es el subprograma principal del proceso iterativo del método desacoplado rápido.

```
CALL SOLFLU (NNODOS, TIPO, TOLP, TOLQ, PG, QG, NMAXIT, IERR, NEXT, VIM,
*          VOLT, ANGULO, INIC, ICOL, YBUSR, YBUSI, ISP, PCAL, QCAL,
*          NPV, NNGEN, DNOMB, QGMIN, QGMAX, NCEVS, PC, QC, NORDEN,
*          JINIC, JICOL, JISP, IFACT, VRE, JL, VESP, KISLA, NISLAS,
*          INISLA, ISLA, NAU, ISLACK, YM, NLT, LONG, TOLREV, BPRIMA,
*          BBPRIM, NSHNTS, NSHUNT, SHUNT, KFUERA, NODFUE)
```

Si el proceso iterativo de la isla No. KISLA diverge, IERR = 1, y no se crean reportes de resultados para la misma.

```
IF(IERR.EQ.1) ISLACK (KISLA) = - 1
```

C

```
30 CONTINUE
```

C

Si IND=0, se define como 'default' la opción 2 del menú principal de flujos. De otro modo la opción de 'default' es 7.

```
IND = 0
```

Se inicializan las variables que utiliza el subprograma MENU.

```
35 CALL INIMEN
```

Se definen las 7 opciones del menú principal de flujos.

```
ENCABE = ' MENU PRINCIPAL DE FLUJOS : '
ALTERN = ' ¿ CUAL OPCION DESEA [1'
LISTOP(1) = ' TERMINAR EL PROGRAMA '
LISTOP(2) = ' REPORTE DE FLUJOS '
LISTOP(3) = ' REPORTE DE PARAMETROS DE LA RED '
LISTOP(4) = ' REPORTES RESUMIDOS '
LISTOP(5) = ' REPORTE PARA ESTABILIDAD '
LISTOP(6) = ' CAMBIOS EN DATOS DEL SISTEMA '
LISTOP(7) = ' OTRO ESTUDIO DE FLUJOS '
```

Se predefine a la variable OPCION con el valor de 2.

```
OPCION = 2
```

Se pregunta si se ha visitado el subprograma de cambios a datos del sistema (TOPCAM).

```
IF (IND.EQ.1) OPCION = 7
```

Se llama al subprograma MENU para definir el valor de OPCION.

```
CALL MENU (OPCION)
IF (OPCION.NE.1) GO TO 130
```

Se llama al subprograma SALIAS, donde se confirma la salida definitiva del simulador. Si INDSAL = 1, se continúa con la simulación.

```
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 35
GO TO 888
130 IF (OPCION.EQ.2) GO TO 140
IF (OPCION.EQ.3) GO TO 160
IF (OPCION.EQ.4) GO TO 180
IF (OPCION.EQ.5) GO TO 180
IF (OPCION.EQ.6) GO TO 210
IF (OPCION.EQ.7) GO TO 300
GO TO 35
```

FINALE es el subprograma de reportes de flujos y pérdidas.

```
140 CALL FINALE (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, YPQ2E, YPQ2R, VRE,
*           VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT, SHUNT, DNOMB,
*           NISLAS, INISLA, ISLA, NAU, NPV, NNGEN, YM, ABORTA, LETR1,
*           LETR2, QGMIN, QGMAX, KFUERA, NODFUE, ISLACK, GPQ2E, GPQ2R)
```

C

```
IF (ABORTA.NE.1) GO TO 35
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 35
GO TO 888
```

DATREP es el subprograma de reportes de parámetros del sistema.

```
160 CALL DATREP (NNODOS, NLT, R, X, R1, X1, GPQ2E, GPQ2R, YPQ2E, YPQ2R,
*           NSHNTS, NSHUNT, SHUNT, DNOMB, NISLAS, INISLA,
*           ISLA, NAU, NPV, NNGEN, YM, ABORTA, LETR1, LETR2, TIPO)
```

C

```
IF (ABORTA.NE.1) GO TO 35
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 35
GO TO 888
```

RESREP es el subprograma de reportes de flujos resumidos.

```
180 CALL RESREP (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, R, X, YPQ2E, YPQ2R,
*           VRE, VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT, SHUNT, DNOMB,
*           NISLAS, INISLA, ISLA, NAU, NPV, NNGEN, YM, ABORTA, LETR1,
*           LETR2, QGMIN, QGMAX, VOLT, ANGULO, ISLACK, TIPGEN, INIC,
*           ICOL, YBUSR, YBUSI, ISP, LONG, OPCION)
```

C

```
IF (ABORTA.NE.1) GO TO 35
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 35
GO TO 888
```

TOPCAM es el subprograma para cambiar parámetros del sistema en estudio

```
210 CALL TOPCAM (YPQ2E, YPQ2R, R, X, NODOS, DNOMB, GPQ2E, GPQ2R, R1, X1, ABORTA,
*           PG, QG, NPV, VESP, QGMIN, QGMAX, NNGEN, TIPGEN, NSHNTS,
*           NSHUNT, SHUNT, PC, QC, NISLAS, ISLA, INISLA, SBASE, TIPO, NAU,
*           LINES, FPCISLA, FQCISLA, NODFUE, KFUERA, LETR1, LETR2)
```

```
IND = 1
IF (ABORTA.NE.1) GO TO 35
CALL SALIAS (INDSAL)
IF (INDSAL.NE.0) GO TO 35
GO TO 888
```

REINIC es el subprograma para reiniciar valores de voltajes y ángulos, así como vectores de inyecciones y tipo de nodos para realizar estudios de flujos posteriores al sistema eléctrico modificado en TOPCAM.

```
300 CALL REINIC (NNODOS, TIPO, PG, QG, NPV, VESP, QGMIN, QGMAX, NNGEN,
*           TIPGEN, NSHNTS, NSHUNT, SHUNT, PC, QC, NISLAS, ISLA,
*           INISLA, FPCISLA, FQCISLA, NAU, PSLACK)
```

NLINES = LINES
 NNODOS = NODOS

Se regresa al subprograma FORMPR.

GO TO 10

Se prepara la salida del simulador a instancia del usuario. Se limpia pantalla con CLEAR y se escribe el mensaje con el formato 999.

888 CALL CLEAR
 WRITE (*, 999)
 999 FORMAT(//20X, 'TERMINA LA SIMULACION POR DECISION DEL USUARIO...')

Se pide presionar una tecla para continuar.

CALL ENCA
 1000 CONTINUE
 CALL CLEAR
 STOP

FIN del programa principal DR.

END

A continuación se describe cada subprograma, siguiendo la clasificación de la Tabla 1, y las ideas de [1].

4. Descripción de Subprogramas para Actividades de Entrada

En esta actividad se efectúa la definición del archivo de datos que se leerá posteriormente, además de parámetros generales del estudio de flujos, así como de los reportes de datos de entrada. Estas actividades se realizan en los subprogramas PORTAD y LEDATO.

a) Subprograma PORTAD

Las actividades de este subprograma son:

1. Definición del archivo de datos del sistema eléctrico a leer.
2. Definición de los parámetros generales del estudio de flujos.
3. Definición de reportes de datos, y sus canales de envío.

La forma de acceder este subprograma es mediante la instrucción:

```
CALL PORTAD (ARCH, ARCH2, ARCH3, LISTIN, PRINTR, ABORTA, SALIDA)
IF (ABORTA.EQ.0) GO TO 5
```

Una descripción de las variables involucradas en el argumento del subprograma, así como del bloque común MENU12 se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ARCH	S	CHARACTER	Nombre del archivo de datos del sistema eléctrico.
ARCH2	S	CHARACTER	Se define en esta variable si el reporte de datos de entrada es por impresora.
ARCH3	S	CHARACTER	Se define en esta variable si el reporte de datos de entrada se almacena en un archivo.
LISTIN	S	ENTERA	Define si existirá reporte de datos de entrada por pantalla.
PRINTR	S	ENTERA	Define si existirá reporte de datos de entrada por archivo y/o impresora.
ABORTA	E/S	ENTERA	Define si el usuario da por terminada la simulación.
SALIDA	E/S	ENTERA	Define si se termina la simulación debido a que el usuario no proporciona correctamente el nombre del del archivo de datos del sistema eléctrico.
VALNUM(15)	E/S	REAL	Almacena números reales que corresponden a variables involucradas en el menú, por ejemplo TOLREV.
DEPEND(15)	E/S	ENTERA	Indica dependencia entre opciones de un menú. Por ejemplo, en el menú de datos generales del estudio de flujos, la opción 8 depende del valor que tiene la opción 7. En este caso, DEPEND(7)=8 y DEPEND(8)=7.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
LISTOP(15)	E	CHARACTER	Contiene las cadenas de caracteres alfanuméricos de cada opción en el menú.
ENCABE	E	CHARACTER	Cadena de caracteres alfanuméricos para identificar cada menú.
ALTERN	E	CHARACTER	Contiene una cadena de caracteres alfanuméricos para formular la pregunta de selección de opción.
TIPVAL(15)	E	CHARACTER	Cadena de tres caracteres alfanuméricos indicando el tipo de variables asociadas con cada opción. Si: TIPVAL(i) = 'REA', opción i asociada a variable real. TIPVAL(i) = 'INT', opción i se asocia a una variable entera. TIPVAL(i) = 'CHA', opción i se asocia a una variable alfanumérica.
VALCHA(15)	E/S	CHARACTER	Contiene variables de tipo alfanumérico asociadas con alguna opción del menú.

Las demás variables utilizadas en este subprograma se describen conforme se explica cada parte de código del mismo, el cual se presenta enseguida.

```

C*****
C   SUBPROGRAMA PORTAD.
C   REALIZADO POR: JOSE HORACIO TOVAR HERNANDEZ.
C                                           OCTUBRE DE 1992.
C*****
C
C   SUBROUTINE PORTAD (ARCH, ARCH2, ARCH3, LISTIN, PRINTR, ABORTA, SALIDA)
C
C   COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
C
C   CHARACTER NAME1*40, NAME2*40, ARCH*40, ARCH2*40, ARCH3*40, ESC*1
C   CHARACTER LISTOP(15)*50, ENCABE*50, ALTERN*50, HOME*3, CONTI*1
C   CHARACTER TIPVAL(15)*3, VALCHA(15)*12, ANSWER*2, RESPUE*2, INVERS*3
C   CHARACTER BLUBCK*4, WHITEFORE*4, YELFORE*4, BLACK*4, ERASEEND*3
C   INTEGER PRINTR, OPCION, ABORTA, SALIDA, DEPEND(15)

```

```

REAL*8      VALNUM(15)
PARAMETER  (YELFORE=' [33m', WHITEFORE=' [37m', ERASEEND=' [K' )
PARAMETER  (BLUBCK=' [44m', INVERS=' [7m', BLACK=' [40m', HOME=' [2J' )

```

Se inicializa el contador KONT para definir el nombre del archivo de datos del sistema eléctrico a leer. Si KONT alcanza el valor de 3, la sesión de simulación finaliza con el mensaje del formato 5011.

```

KONT = 0
ESC=CHAR(27)

```

Se imprime en pantalla la presentación del simulador de flujos.

```

CALL  PANTAL

```

Se lee el nombre del archivo de 'default' de datos del sistema a leer por el subprograma LEDATO. Este nombre está contenido en el archivo ARCHI.FIL, el cual debe existir en el directorio de trabajo.

```

OPEN (UNIT = 9, FILE = 'ARCHI.FIL', STATUS = 'OLD', ERR=40,
*     ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
READ (9,5000) NAME1
CLOSE (UNIT = 9)

```

Se inicializa la variable ARCH con el nombre contenido en ARCHI.FIL

```

ARCH = NAME1

```

Se borra y pinta un marco azul en pantalla en el subprograma CUADRO.

```

29 CALL  CUADRO

```

Se da la oportunidad de leer otro archivo de datos en la siguiente - pregunta.

```

WRITE (*,5020) ESC,ESC,YELFORE,ESC,WHITEFORE,ESC,NAME1,ESC
5020 FORMAT (A,' [5;13H',A,A4,' ARCHIVO DE DATOS ? ',
*          A,A4,/,A,' [5;35H',A40,A,' [5;35H', '$)
READ (*,5000) NAME2
IF (NAME2.NE.' ') NAME1 = NAME2
IF (NAME2.EQ.' ') GO TO 38

```

Se intenta abrir el archivo con el nombre leído y definido por NAME1. Si se tiene éxito se pasa a la etiqueta 38. Si existe error a la 35.

```

OPEN (UNIT = 9, FILE = NAME2      , STATUS = 'OLD', ERR=35,
*     ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
CLOSE (UNIT = 9)
GO TO 38

```

Se reinicia el procedimiento para intentar definir el nombre de archivo Si KONT alcanza el valor de 3 (intentos) la simulación se finaliza.

```

35 NAME1 = ARCH
WRITE (*,5010) ESC, NAME2
KONT = KONT + 1
CALL ENCA

```

```

IF (KONT.LT.4) GO TO 29
WRITE (*,5011) ESC
SALIDA = 1
RETURN

```

En esta secuencia se observa si el nombre de archivo de 'default' es el mismo que el dado por el usuario. Si son distintos, entonces, se da la oportunidad de definir a este último como nuevo archivo de 'default'.

```

38 IF (ARCH.EQ.NAME1) GO TO 50
WRITE (*,4925) ESC, YELFORE, ESC, ESC
WRITE (*,4926) ESC, WHITEFORE
WRITE (*,4927) ESC, ARCH
WRITE (*,4928) ESC, YELFORE, ESC
WRITE (*,4926) ESC, WHITEFORE
WRITE (*,4929) ESC, NAME1
WRITE (*,4930) ESC, YELFORE, ESC, ESC, WHITEFORE, ESC
READ (*,4940) ANSWER
IF (ANSWER.NE.' '.AND.ANSWER.NE.'SI'.AND.ANSWER.NE.'si')GO TO 50

```

Aquí, se escribe el nombre del nuevo archivo de 'default'.

```

OPEN (UNIT = 9, FILE = 'ARCHI.FIL', STATUS = 'OLD', ERR=40,
*      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
WRITE(9,5000) NAME1
CLOSE (UNIT = 9)
GO TO 50

```

Se define secuencia de salida por error en la apertura de ARCHI.FIL.

```

40 WRITE (*,5009) ESC
SALIDA = 1
RETURN

```

Se define definitivamente el nombre del archivo de datos a leer.

```

50 ARCH = NAME1

```

Aquí, se inicializan variables y nombres de archivos para reportes.

```

LISTIN = 0
PRINTR = 0
ARCH2 = ' '
ARCH3 = ' '

```

Se inicializan las variables para el subprograma MENU.

```

CALL INIMEN

```

Se definen las opciones para el menú de reportes de datos de entrada.

```

ENCABE = ' OPCIONES PARA DATOS DE ENTRADA : '
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' CONTINUAR EL PROGRAMA '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' LISTADO DE DATOS POR PANTALLA '
LISTOP(4) = ' LISTADO DE DATOS POR IMPRESORA '

```

```
LISTOP(5) = ' GRABAR LISTADO DE DATOS EN ARCHIVO '
51 OPCION = 1
```

Se selecciona una de las opciones anteriores en el subprograma MENU.

```
CALL MENU (OPCION)
IF (OPCION.EQ.1) GO TO 55
IF (OPCION.NE.2) GO TO 52
ABORTA = 1
RETURN
52 IF (OPCION.NE.3) GO TO 53
```

Se define la opción de reporte de datos por pantalla.

```
LISTIN = 1
GO TO 51
53 IF (OPCION.NE.4) GO TO 54
IF (PRINTR.LT.2) PRINTR = PRINTR + 1
```

Se define la opción de reporte de datos por impresora.

```
ARCH2 = 'PRN'
GO TO 51
```

Se define la opción de grabar el reporte de datos en archivo. El nombre del archivo de 'default' es DATOS.SAL.

```
54 NAME1 = 'DATOS.SAL'
IF (PRINTR.LT.2) PRINTR = PRINTR + 1
ARCH3 = NAME1
CALL CUADRO
WRITE (*,5120)ESC,ESC,YELFORE,ESC,WHITEFORE,ESC,NAME1,ESC
5120 FORMAT (A,'[5;13H',A,A4,' NOMBRE DEL ARCHIVO ? ',
*          A,A4,/,A,'[5;35H',A40,A,'[5;35H',$,)
READ (*,5000) NAME2
IF (NAME2.EQ.' '.OR.NAME2.EQ.NAME1) GO TO 51
ARCH3 = NAME2
GO TO 51
55 WRITE(*,5) ESC, BLACK, ESC, HOME, ESC, YELFORE
```

Se abre el archivo ESTUDIO.PAR que contiene los datos generales para el estudio de flujos:

```
TOLP = tolerancia máxima de error en desviaciones de potencia real.
TOLQ = tolerancia máxima de error en desviaciones de potencia reactiva.
NMAXIT = número máximo de iteraciones del proceso de solución.
SBASE = potencia base del sistema eléctrico en MVA.
TINIA = tolerancia máxima de error para revisar límites de potencia -
reactiva de generación.
ANSWER = esta variable tiene valor de 'SI' o 'NO'. En el primer caso,
se revisan límites de Q generada. En el último, no se revisan.
```

```
OPEN (UNIT = 9, FILE = 'ESTUDIO.PAR', STATUS = 'OLD', ERR=40,
*      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
READ (9,5100) TOLP, TOLQ, NMAXIT, SBASE, ANSWER, TINIA
5100 FORMAT(F8.6,2X,F8.6,2X,I3,2X,F8.4,3X,A2,3X,F8.6)
CLOSE (UNIT = 9)
```

Iniciación de variables a ser usadas en MENU, la cual consiste en definir las variables asociadas a cada opción como alfanuméricas.

CALL INIMEN

Se definen variables asociadas a las opciones 3, 4 y 5 como tipo REAL.
 Se define la variable asociada a la opción 6 como ENTERA.
 Se define a la variable asociada a la opción 8 como tipo REAL.
 Se define la opción 8 como dependiente de la opción 7.
 Se inicializan todas las variables con los datos leídos de ESTUDIO.PAR.

```
TIPVAL(3) = 'REA'
TIPVAL(4) = 'REA'
TIPVAL(5) = 'REA'
TIPVAL(6) = 'INT'
TIPVAL(8) = 'REA'
VALNUM(3) = TOLP
VALNUM(4) = TOLQ
VALNUM(5) = SBASE
VALNUM(6) = NMAXIT
VALNUM(8) = TINIA
DEPEND(7) = 8
VALCHA(7) = 'SI'
DEPEND(8) = 7
IF (ANSWER.EQ.'SI') GO TO 57
VALCHA(7) = 'NO'
DEPEND(8) = -7
```

Se definen las opciones para el menú de modificación de datos generales del estudio de flujos, los cuales están almacenados en ESTUDIO.PAR

```
57 ENCABE = ' DATOS GENERALES DEL ESTUDIO : '
ALTERN = ' ( CUAL OPCION DESEA [ 1'
LISTOP(1) = ' CONTINUAR EL PROGRAMA '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' TOLERANCIA EN LA POTENCIA REAL'
LISTOP(4) = ' TOLERANCIA EN LA POTENCIA REACTIVA'
LISTOP(5) = ' POTENCIA BASE DEL SISTEMA (MVA)'
LISTOP(6) = ' NUMERO MAXIMO DE ITERACIONES'
LISTOP(7) = ' SE CONSIDERAN LIMITES EN Qg'
LISTOP(8) = ' MAX. DESV. PARA REVISAR LIMITES EN Qg'
OPCION = 1
CALL MENU (OPCION)
IF (OPCION.NE.2) GO TO 60
ABORTA = 1
RETURN
60 IND = 5
```

Se actualiza la variable ANSWER.

```
IF (VALCHA(7).EQ.'SI'.OR.VALCHA(7).EQ.'sI'.OR.
* VALCHA(7).EQ.'si'.OR.VALCHA(7).EQ.'Si') ANSWER = 'SI'
IF (VALCHA(7).EQ.'NO'.OR.VALCHA(7).EQ.'no'.OR.
* VALCHA(7).EQ.'no'.OR.VALCHA(7).EQ.'No') ANSWER = 'NO'
```

Se actualiza la variable TOLP. Se revisa el rango de valores aceptable.

```
IF (VALNUM(3).LT.0.00006.OR.VALNUM(3).GT.0.1) GO TO 62
TOLP = VALNUM(3)
IND = IND - 1
```

Se actualiza la variable TOLQ. Se revisa el rango de valores aceptable.

```
62 IF (VALNUM(4).LT.0.00006.OR.VALNUM(4).GT.0.1) GO TO 63
TOLQ = VALNUM(4)
IND = IND - 1
```

Se actualiza la variable SBASE. Se revisa rango de valores aceptable.

```
63 IF (VALNUM(5).LT.10.0.OR.VALNUM(5).GT.10000.0)GO TO 64
SBASE = VALNUM(5)
IND = IND - 1
```

Se actualiza la variable NMAXIT. Se revisa rango de valores aceptable.

```
64 IF (VALNUM(6).LT.15.0.OR.VALNUM(6).GT.200.0) GO TO 65
NMAXIT = INT(VALNUM(6))
IND = IND - 1
```

Se actualiza la variable TINIA. Se revisa rango de valores aceptable.

```
65 IF (VALNUM(8).LT.0.00006.OR.VALNUM(8).GT.0.1) GO TO 66
TINIA = VALNUM(8)
IND = IND - 1
```

Se verifica si alguna de todas las variables estuvo fuera de rango.

```
66 IF (IND.EQ.0) GO TO 80
CALL CUADRO
WRITE (*,68) ESC, ESC, YELFORE, ESC
68 FORMAT (A, '[5;5H', A, A4,
* 'SE LEYERON ALGUNOS DATOS FUERA DE RANGO.', /, A, '[5;47H',
* 'SE CONSIDERARAN LOS SIGUIENTES:')
WRITE (*,70) ESC, ESC, ESC, WHITEFORE, TOLP, ESC, YELFORE
70 FORMAT(A, '[8;10H', 'MAXIMA DESVIACION DE POTENCIA REAL : ',
* /, A, '[8;50H', A, A4, F10.6, A, A4)
WRITE (*,71) ESC, ESC, ESC, WHITEFORE, TOLQ, ESC, YELFORE
71 FORMAT(A, '[10;10H', 'MAXIMA DESVIACION DE POT. REACTIVA : ',
* /, A, '[10;50H', A, A4, F10.6, A, A4)
WRITE (*,72) ESC, ESC, ESC, WHITEFORE, SBASE, ESC, YELFORE
72 FORMAT(A, '[12;10H', 'POTENCIA BASE DEL SISTEMA (MVAs) : ',
* /, A, '[12;50H', A, A4, F10.6, A, A4)
WRITE (*,73) ESC, ESC, ESC, WHITEFORE, NMAXIT, ESC, YELFORE
73 FORMAT(A, '[14;10H', 'MAXIMO NUMERO DE ITERACIONES : ',
* /, A, '[14;50H', A, A4, I3, A, A4)
WRITE (*,74) ESC, ESC, ESC, WHITEFORE, ANSWER, ESC, YELFORE
74 FORMAT(A, '[16;10H', 'REVISION DE LIMITES EN Qg : ',
* /, A, '[16;50H', A, A4, A2, A, A4)
IF (ANSWER.EQ.'NO') GO TO 77
WRITE (*,76) ESC, ESC, ESC, WHITEFORE, TINIA, ESC, YELFORE
```

```

76 FORMAT(A, '[18;10H', 'MAX. DESV. PARA REVISAR LIMITES EN Qg : ',
*      /, A, '[18;50H', A, A4, F10.6, A, A4)
77 WRITE(*, 78) ESC
78 FORMAT(A, '[22;20',
*      ', Oprima la tecla de "entrada" para continuar...')
  READ (*, 79) OPCION
79 FORMAT(I2)

```

Se graba el archivo ESTUDIO.PAR con los nuevos valores de datos.

```

80 OPEN (UNIT = 9, FILE = 'ESTUDIO.PAR', STATUS = 'OLD', ERR=40,
*      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
  WRITE (9,5100) TOLP, TOLQ, NMAXIT, SBASE, ANSWER, TINIA
  CLOSE (UNIT = 9)
90 CONTINUE

```

Formatos de escritura.

```

4903 FORMAT (A, '[', I2.2, ';1H', A, A4, A, A3)
4904 FORMAT (A, '[', I2.2, ';2H', A, A4, A, A3)
4905 FORMAT (A, '[', I2.2, ';80H', A, A4, A, A3)
4917 FORMAT (A, '[22;', I2.2, 'H', A, A4, A, A3)
4925 FORMAT (A, A4, A, '[15;3H',
*      '( SE CAMBIA EL NOMBRE DEL ARCHIVO DE DEFAULT', /,
*      A, '[16;3H', '----->')
4926 FORMAT (A, A4)
4927 FORMAT (A, '[16;39H', A40)
4928 FORMAT (A, A4, A, '[18;3H', 'POR EL DEL ARCHIVO SELECCIONADO -->')
4929 FORMAT (A, '[18;39H', A40)
4930 FORMAT (A, A4, A, '[20;30H', '[SI, NO] ?', A, A4, ' SI', A, '[20;41H', '$)
4940 FORMAT (A2)
4957 FORMAT (A, A4, A, '[18;47H', '1', A, '[18;47H', '$)
4958 FORMAT (A1)
5000 FORMAT (A40)
5007 FORMAT (A, A4, A, '[22;1', '')
5009 FORMAT (A, '[18;25H', 'ERROR EN LA APERTURA DE "ARCHI.FIL" ')
5010 FORMAT (A, '[18;10H', 'EL ARCHIVO ', A40, ' NO EXISTE !!! ')
5011 FORMAT (A, '[19;15H', 'REVISE EL DIRECTORIO ACTUAL !!! ')
5030 FORMAT (A, '[', I2.2, ';70H', A, A, '$)
5040 FORMAT (I1)
5050 FORMAT (F7.5)
5055 FORMAT (F7.2)
5060 FORMAT (I3)

```

Fin del subprograma PORTAD.

```

RETURN
END

```

b) Subprograma LEDATO

Las actividades de este subprograma son:

1. Lectura de datos del sistema eléctrico de potencia.
2. Lectura de los parámetros generales del estudio definidos en el subprograma PORTAD.
3. Impresión de reportes de datos de entrada, previamente definidos en PORTAD.

La forma de acceder este subprograma es mediante la instrucción:

```
CALL LEDATO (ARCH, DNOMB, LETR1, LETR2, NE, NR, NPV, NAU, NSHUNT, R, X, YPQ2E,
*          PC, QC, PG, VESP, QGMAX, QGMIN, SHUNT, QG, NNODOS, NLINES,
*          NNGEN, NSHNTS, NCEVS, LISTIN, PRINTR, TIPGEN, ARCH2, ARCH3,
*          IERR, TIPO, TOLP, TOLQ, NMAXIT, SBASE, TOLREV)
```

Una descripción de las variables involucradas en el argumento del subprograma y del bloque común TRANSF se muestran a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ARCH	E	CHARACTER	Nombre del archivo de datos del sistema eléctrico.
ARCH2	E	CHARACTER	Se define en esta variable si el reporte de datos de entrada es por impresora.
ARCH3	E	CHARACTER	Se define en esta variable si el reporte de datos de entrada se almacena en un archivo.
LISTIN	E	ENTERA	Define si existirá reporte de datos de entrada por pantalla.
PRINTR	E	ENTERA	Define si existirá reporte de datos de entrada por archivo y/o impresora.
NNODOS	S	ENTERA	Número total de nodos del sistema eléctrico.
DNOMB(NNODOS)	S	CHARACTER	Nombre de cada nodo en el sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
LETR1	S	CHARACTER	Primer renglón descriptivo del archivo de datos del sistema eléctrico.
LETR2	S	CHARACTER	Segundo renglón descriptivo del archivo de datos.
IERR	E/S	ENTERA	IERR = 0 cuando no existen errores en el proceso de lectura de datos. De otro modo, IERR = 1, avisando al programa principal de tales anomalías.
PC(NNODOS)	S	REAL	Potencia real de carga en cada nodo del sistema eléctrico de potencia.
QC(NNODOS)	S	REAL	Potencia reactiva de carga en cada nodo del sistema eléctrico de potencia.
PG(NNODOS)	S	REAL	Potencia real de generación en cada nodo del sistema eléctrico de potencia.
QG(NNODOS)	S	REAL	Potencia reactiva de generación en cada nodo del sistema eléctrico.
TIPO(NNODOS)	S	ENTERA	Tipo de cada nodo del sistema eléctrico de potencia. Si: TIPO(i) = 1, i es nodo compensador. TIPO(i) = 2, i es nodo de voltaje controlado. TIPO(i) = 3, i es nodo de carga.
NNGEN	S	ENTERA	Número total de nodos de generación (nodos tipo voltaje controlado).
NPV(NNGEN)	S	ENTERA	Contiene el identificador (número entero) de cada nodo de generación.
NAU(NNGEN)	S	ENTERA	Contiene el identificador del nodo de alta tensión asociado a cada generador.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
VESP(NNGEN)	S	REAL	Voltaje especificado en terminales del generador.
QGMAX(NNGEN)	S	REAL	Limite máximo de potencia reactiva generada.
QGMIN(NNGEN)	S	REAL	Limite mínimo de potencia reactiva generada.
TIPGEN(NNGEN)	S	ENTERA	Tipo de generador: 0 para máquina síncrona, 1 para compensador estático de potencia reactiva.
NCEVS	S	ENTERA	Número total de compensadores estáticos de potencia reactiva.
NLINES	S	ENTERA	Número total de elementos de transmisión en el sistema eléctrico de potencia.
NE(NLINES)	S	ENTERA	Identificador del nodo de envío de cada elemento de transmisión.
NR(NLINES)	S	ENTERA	Identificador del nodo de recepción de cada elemento de transmisión.
R(NLINES)	S	REAL	Resistencia serie de cada elemento de transmisión en valores por unidad.
X(NLINES)	S	REAL	Reactancia serie de cada elemento de transmisión en valores por unidad.
YPQ2E(NLINES)	S	REAL	Parte imaginaria de la mitad del efecto capacitivo de cada elemento de transmisión.
NSHNTS	S	ENTERA	Número total de elementos de compensación fija en derivación.
NSHUNT(NSHNTS)	S	ENTERA	Identificador de cada nodo en el que existe un compensador en derivación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
SHUNT(NSHNTS)	S	REAL	Valor de la susceptancia en por unidad de cada compensador en derivación.
TOLP	S	REAL	Tolerancia de error máximo para desviaciones de potencia real.
TOLQ	S	REAL	Tolerancia de error máximo para desviaciones de potencia reactiva.
NMAXIT	S	ENTERA	Máximo de iteraciones que puede tener el proceso de solución.
TOLREV	S	REAL	Tolerancia de error máximo en desviaciones de potencia reactiva para iniciar la revisión de límites de generación.
SBASE	S	REAL	Potencia base en MVA del sistema eléctrico de potencia.
NTAPS	S	ENTERA	Total de transformadores con tap variable.
NETR(NTAPS)	S	ENTERA	Identificador del nodo al lado del tap de transformadores con tap variable.
NRTR(NTAPS)	S	ENTERA	Identificador del nodo al lado contrario del tap de transformadores con tap variable.
TAP(NTAPS)	S	REAL	Valor del tap en cada uno de los transformadores con tap variable.

La descripción de las demás variables utilizadas por este subprograma se presenta conforme se avanza en la explicación de cada parte de código, el cual se muestra a continuación.

```

C*****
C****   LEDATO ES UN PROGRAMA DE LECTURA DE DATOS PARA
C****   ESTUDIOS DE FLUJOS DE POTENCIA
C****   AUTOR: RICARDO MOTA PALOMINO   SEPTIEMBRE DE 1989.
C****   MODIFICACIONES A ESTA VERSION: JOSE HORACIO TOVAR HERNANDEZ.
C****                                     OCTUBRE DE 1992.
C*****

```

```

C
C   SUBROUTINE LEDATO (ARCH, DNOMB, LETR1, LETR2, NE, NR, NPV, NAU, NSHUNT, R, X,
*   YPQ2E, PC, QC, PG, VESP, QGMAX, QGMIN, SHUNT, QG, NNODOS,
*   NLINES, NNGEN, NSHNTS, NCEVS, LISTIN, PRINTR, TIPGEN,
*   ARCH2, ARCH3, IERR, TIPO, TOLP, TOLQ, NMAXIT, SBASE,
*   TOLREV)

```

Declaración de variables.

```

C   COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
C
C   INTEGER   NE(1000), NR(1000), NETR(300), NRTR(300), TIPO(550),
*   NPV(150), NAU(150), NSHUNT(150), PRINTR, TIPGEN(150)
C
C   REAL     R(1000), X(1000), YPQ2E(1000), PC(550), QC(550), TAP(300), SBASE,
*   VESP(150), QGMAX(150), QGMIN(150), PG(550), SHUNT(150), QG(550)
C
C   CHARACTER ARCH1*20, ARCH2*40, ARCH3*40, ARCH21*40, ARCH31*40,
*   NAME*40, LETR1*70, LETR2*70, NOMB1*8, NOMB2*8,
*   IFIN*8, DNOMB(550)*8, IRES*8, IRE1*8, ANSWER*2,
*   ESC*1, BLUBCK*4, WHITEFORE*4, YELFORE*4, ARCH*40

```

Se inicializan las variables para crear efectos de pantalla.

```

C   PARAMETER (YELFORE=' [33m', WHITEFORE=' [37m', BLUBCK=' [44m' )
C
C   LOGICAL   IEX

```

Se inicializan variables auxiliares para lectura de datos.

```

KONT = indica el renglón de pantalla donde se posiciona el cursor.
LECTU = canal de lectura en forma interactiva.
LTERM = canal de escritura por pantalla.
LFFLU = canal de lectura del archivo de datos del sistema eléctrico.
ITIP = auxiliar para revisar nombres de nodos en el subprograma -
VERIF. Si ITIP = 1, se revisan dos nombres. Si ITIP = 2, se
revisa un solo nombre.

```

```

KONT   = 7
LECTU  = 5
LTERM  = 6
IFIN   = '0'
IWR    = LTERM
ESC    = CHAR(27)
LFFLU  = 9

```

```

NNODOS = 0
NLINEAS = 0
NNGEN = 0
NCEVS = 0
NTAPS = 0
NSHNTS = 0
ITIP = 1
IERR = 0

```

Se abre el archivo ESTUDIO.PAR para leer los parámetros generales del estudio de flujos.

```

OPEN (UNIT = 9, FILE = 'ESTUDIO.PAR', STATUS = 'OLD', ERR=40,
*     ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
READ  (9, 5) TOLP, TOLQ, NMAXIT, SBASE, ANSWER, TOLREV
CLOSE (UNIT = 9)
5  FORMAT (F8.6, 2X, F8.6, 2X, I3, 2X, F8.4, 3X, A2, 3X, F8.6)
40 CONTINUE

```

Se modifica TOLREV en caso de que no se haya especificado revisar - límites de potencia reactiva generada, tal que sea menor que la - tolerancia de error máximo en desviaciones de potencia reactiva.

```

IF (ANSWER.EQ.'NO') TOLREV = TOLQ / 100.0

```

```

C
C **** NNODOS ES EL NUMERO TOTAL DE NODOS
C **** NLINEAS ES EL NUMERO TOTAL DE LINEAS
C **** NNGEN ES EL NUMERO TOTAL DE NODOS DE GENERACION
C **** NTAPS ES NUMERO TOTAL DE TAPS FUERA DEL NOMINAL EN TRANSFORMADORES
C **** NSHNTS NUMERO TOTAL DE COMPENSADORES EN PARALELO
C **** NCEVS NUMERO DE COMPENSADORES ESTATICOS DE POT. REACTIVA
C
C ***** INICIA LECTURA DE DATOS. SE ABRE EL ARCHIVO DE DATOS.
C
C     OPEN (UNIT = 9, FILE = ARCH, STATUS = 'OLD',
*         ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
C

```

Si se especificó que se desea un archivo con datos de entrada o - impresión de los mismos por impresora, se especifica el canal de - escritura.

```

IF (PRINTR.EQ.1) IWR = 8
CALL CUADRO

```

Se lee el primer renglón descriptivo del archivo de datos del sistema.

```

READ (LFFLU, 43, ERR = 48) LETR1

```

Se escribe este primer renglón en pantalla.

```

WRITE (6,44) ESC, ESC, WHITEFORE, LETR1

```

Se escribe este primer renglón en impresora.

```

IF (IWR.EQ.8) WRITE(IWR,43) LETR1

```

```

43  FORMAT (1X,A70)
44  FORMAT (A,'[2;5H', A, A4, A68)
45  FORMAT (A,'[3;5H', A, A4, A68)

```

Se lee el segundo renglón descriptivo del archivo de datos del sistema.

```
READ (LFFLU, 43, ERR = 48) LETR2
```

Se escribe este segundo renglón en pantalla.

```
WRITE (6,45) ESC, ESC, WHITEFORE, LETR2
```

Se escribe este segundo renglón en impresora.

```
IF(IWR.EQ.8) WRITE (IWR,43) LETR2
GO TO 54
```

Se escribe mensaje de error de lectura. Se activa IERR y se regresa al programa principal.

```

48  WRITE(6, 50) ESC
50  FORMAT(A,'[10;10H', '***** ERROR EN LA DESCRIPCION DEL CASO ')
    IERR = 1
    GO TO 3000

```

C

C ***** LECTURA DE INFORMACION NODAL

C

Se inicia contador de nodos en el sistema eléctrico de potencia.

```
54  I = 1
```

Se lee nombre de nodo, potencia real y potencia reactiva de carga.

```
58  READ (LFFLU, 60, ERR = 66) NOMB1, PC(I), QC(I)
```

Se verifica si es el final de este bloque de datos.

```

    IF (NOMB1.EQ.IFIN) GO TO 80
60  FORMAT (2X,A8,2X,F8.2,2X,F8.2)

```

Se guarda el nombre del nodo en la variable DNOMB(I).

```
DNOMB(I) = NOMB1
```

Se actualiza el valor de la variable NNODOS que indica el total de nodos leídos correctamente en el sistema.

```

NNODOS = NNODOS + 1
I = I + 1

```

Se regresa a leer otro nodo, potencia real y potencia reactiva de carga

```
GO TO 58
```

Secuencia para informar error en la lectura de datos de este bloque.

```

66  WRITE(6,69)
69  FORMAT (//5X,'*** ERROR EN LA INFORMACION NODAL')

```

```
IERR = 1
GO TO 3000
```

Se escribe mensaje de los nodos leídos.

```
80 WRITE (6,82) ESC, NNODOS
82 FORMAT(A,'[8;15H','SE LEE LA INFORMACION DE ',I6,' NODOS')
C
C SE INICIALIZA EL TIPO DE NODOS
C
C SI TIPO(I) = 1, EL NODO ES COMPENSADOR,
C SI TIPO(I) = 2, EL NODO ES GENERADOR,
C SI TIPO(I) = 3, EL NODO ES DE CARGA.
C
      DO 84 I = 1, NNODOS
          TIPO(I) = 3
          PG(I) = 0.0
          QG(I) = 0.0
      84 CONTINUE
C
C ***** INICIO DE LECTURA DE GENERADORES
C
```

Se inicia contador de nodos de generación del sistema eléctrico.

```
I = 1
```

Se lee : nodo de generación, nodo de alta tensión asociado,
potencia real de generación,
límite máximo de potencia reactiva de generación,
límite mínimo de potencia reactiva de generación,
voltaje especificado en terminales del generador,
tipo de generador (máquina síncrona o CEV).

```
88 READ (LFFLU,94, ERR = 105) NOMB1, NOMB2, PGEN , QGMAX(I),
*                               QGMIN(I), VESP(I), TIPGEN(I)
94 FORMAT (2X,A8,2X,A8,2X,3(F8.2,2X),F8.6,1X,I1)
```

Se verifica si es el final de este bloque de datos.

```
97 IF (NOMB1.EQ.IFIN) GO TO 116
```

Se actualiza la variable para el total de nodos de generación.

```
NNGEN = NNGEN + 1
I = I + 1
```

Se verifica que los nombres de nodo NOMB1 y NOMB2 estén en la lista de nodos almacenada en DNOMB.

```
CALL VERIF (NNODOS, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
```

Se asignan identificadores de estos nombres de nodos a los arreglos NPV y NAU, respectivamente. Además, se guarda en TIPGEN el tipo de nodo.

```
NPV(NNGEN) = JX
NAU(NNGEN) = JY
```

Se identifica al nodo JX de generación como tipo PV.

```
TIPO(JX) = 2
PG(JX) = PGEN
```

Si existen nombres de nodos no registrados la información anterior se elimina.

```
IF (.NOT.IEX) NNGEN = NNGEN - 1
IF (.NOT.IEX) GO TO 88
```

Si TIPGEN(NNGEN) = 0 se trata de una máquina síncrona.

```
IF (TIPGEN(NNGEN).EQ.0) GO TO 88
```

Se inicializa información de nodos con compensadores estáticos de VARs.

```
NCEVS = NCEVS + 1
NAU(NNGEN) = - JY
GO TO 88
```

Secuencia de mensaje para error en lectura de este bloque de datos.

```
105 WRITE (6,108) NNGEN
108 FORMAT(/5X,'***** ERROR EN DATOS DE GENERACION',/5X,
* 'EN LA LINEA NUMERO ',I6)
IERR = 1
GO TO 3000
```

Se envía mensaje de lo que se leyó con respecto a nodos PV.

```
116 NL = NNGEN - NCEVS
WRITE(6,119) ESC, NL, ESC, NCEVS
119 FORMAT(A,'[11;15H',' SE LEYO INFORMACION DE ',I3,
* ' NODOS DE GENERACION',/,
* A,'[12;15H',' Y ',I3,' COMPENSADORES ESTATICOS DE VARS')
```

C

C ***** INICIO DE LA LECTURA DE DATOS DE LINEAS Y TRANSFORMADORES

C

Se inicializa contador de elementos de transmisión.

```
I = 1
```

Se lee información de elementos de transmisión:

nombre de nodo de envío, nombre de nodo de recepción, resistencia serie
reactancia serie,
mitad de la parte imaginaria de la susceptancia en derivación,
valor de tap (TAP = 0, cuando se trata de una línea de transmisión o un
transformador con tap nominal y fijo).

```
140 READ(LFFLU, 142, ERR = 163) NOMB1,NOMB2,R(I),X(I),YPQ2E(I),T
142 FORMAT (2X,A8,2X,A8,2X,4(F7.5,2X),I1)
```

Se verifica si es el final de este bloque de datos.

```
IF (NOMB1.EQ.IFIN)GO TO 175
```


Se actualiza la variable con el total de elementos de transmisión.

```
NLINES = NLINES + 1
I = I + 1
```

Se verifica que los nombres de nodo NOMB1 y NOMB2 estén en la lista de nodos almacenada en DNOMB.

```
CALL VERIF (NNODOS, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
```

Si existe algún nombre de nodo no registrado en DNOMB(.), se elimina la información leída.

```
IF (.NOT.IEX) NLINES = NLINES - 1
IF (.NOT.IEX) GO TO 140
```

Se asignan los identificadores JX y JY a las variables que almacenan los nodos de envío y recepción de cada elemento de transmisión.

```
NE (NLINES) = JX
NR (NLINES) = JY
```

Se verifica si se trata de un tap fijo o variable.

```
IF (T.LE.1.0E-8) GO TO 140
```

Se actualiza información de transformadores con tap variable.

```
NTAPS = NTAPS + 1
NETR(NTAPS) = JX
NRTR(NTAPS) = JY
TAP (NTAPS) = T
GO TO 140
```

Secuencia de mensaje de error de lectura en este bloque de datos.

```
163 WRITE (6,168) NLINES
168 FORMAT(//5X, ' ERROR EN LOS DATOS DE LINEA ',/5X,
* ' EN LA LINEA NUMERO ',I6)
IERR = 1
GOTO 3000
```

Secuencia de mensaje de los elementos de transmisión leídos.

```
175 NL = NLINES - NTAPS
WRITE (6,179) ESC, NL, ESC, NTAPS
179 FORMAT(A, '[14;15H', 'LEIMOS LA INFORMACION DE', I6, ' LINEAS Y ',/,
* A, '[15;30H', I6, ' TRANSFORMADORES ')
```

```
C
C ***** INICIO DE LA LECTURA DE ELEMENTOS 'SHUNT'
C
```

Se inicializa contador de compensadores en derivación fijos.

```
I = 1
```

Se asigna valor a ITIP tal que el subprograma VERIF revise únicamente el nombre de un nodo.

```
ITIP = 2
```

Se lee el nombre de nodo , valor de la susceptancia del compensador.

```
250 READ (LFFLU, 252, ERR = 253) NOMB1, SHUNT(1)
252 FORMAT (2X, A8, 2X, F8.5)
```

Se verifica si es el final de este bloque de información.

```
IF (NOMB1.EQ.IFIN) GO TO 266
```

Se actualiza la variable NSHNTS que contiene el total de compensadores en derivación.

```
NSHNTS = NSHNTS + 1
I = I + 1
```

Se revisa el nombre de nodo leído.

```
CALL VERIF (NNODOS, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
```

Si es nombre de nodo no registrado, la información leída se elimina.

```
IF (IEX) NSHUNT(NSHNTS) = JX
IF (.NOT.IEX) NSHNTS = NSHNTS - 1
GO TO 250
```

Secuencia de mensaje de error en lectura de este bloque de datos.

```
253 WRITE (6, 258) ESC, ESC, NSHNTS
258 FORMAT (A, '[19;15H', '**** ERROR EN LOS DATOS DE SHUNTS ', '/',
* A, '[20;15H', ' EN LA LINEA NUMERO ', I6)
IERR = 1
GO TO 3000
```

Se escribe mensaje de los nodos con compensación fija leídos.

```
266 WRITE (6, 269) ESC, NSHNTS
269 FORMAT (A, '[19;15H', 'SE LEYO INFORMACION DE ', I6, ' SHUNTS ')
```

Se cierra el archivo de datos del sistema eléctrico de potencia.

```
CLOSE (UNIT = 9)
```

Se inicia el proceso de impresión de reportes de datos de entrada.

```
CALL ENCA
```

Si la condición siguiente se cumple, se pasa a reportes por impresora o archivo.

```
IF (LISTIN.NE.1) GO TO 600
```

Se imprime reporte de datos por pantalla.

```

KONT = 7
CALL CUADRO
WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
WRITE (*, 276) ESC
276  FORMAT (A, '[5;20H',
*      '      ' NODO ', 2X, 'P-CARGA (MW)', 2X, 'Q-CARGA (MVAR)')
DO 300 I = 1, NNODOS
  IF (KONT.LT.23) GO TO 279
  KONT = 7
  CALL ENCA
  CALL CUADRO
  WRITE (*, 276) ESC
  WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
  WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
279  IF (KONT.LT.10) WRITE(*, 280) ESC, KONT, DNOMB(I), PC(I), QC(I)
  IF (KONT.GE.10) WRITE(*, 283) ESC, KONT, DNOMB(I), PC(I), QC(I)
280  FORMAT (A, '[' , I1, ' ; 20H' , A8, 3X, F8.2, 5X, F8.2)
283  FORMAT (A, '[' , I2, ' ; 20H' , A8, 3X, F8.2, 5X, F8.2)
      KONT = KONT + 1
300  CONTINUE
  CALL ENCA
  KONT = 7
  CALL CUADRO
  WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
  WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
  WRITE (*, 306) ESC, ESC, ESC
306  FORMAT (A, '[5;4H', ' NODO-GEN', ' NODO-ALTA', 2X, 'P-GEN (MW)', /,
*      A, '[5;37H', ' QG-MAX(pu)', 2X, 'QG-MIN(pu)', /,
*      A, '[5;61H', ' V-GEN (pu)', 2X, 'TIPO')
DO 350 I = 1, NNGEN
  NAME = 'GEN'
  IF (TIPGEN(I).EQ.1) NAME = 'CEV'
  JX = NPV(I)
  JY = NAU(I)
  JY = ABS(JY)
  NOMB1 = DNOMB(JX)
  NOMB2 = DNOMB(JY)
  IF (KONT.LT.23) GO TO 330
  KONT = 7
  CALL ENCA
  CALL CUADRO
  WRITE (*, 306) ESC, ESC, ESC
  WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
  WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
330  IF (KONT.LT.10) WRITE (*, 340) ESC, KONT, NOMB1, NOMB2, PG(JX),
*      QGMAX(I), QGMIN(I), VESP(I), NAME
  IF (KONT.GE.10) WRITE (*, 345) ESC, KONT, NOMB1, NOMB2, PG(JX),
*      QGMAX(I), QGMIN(I), VESP(I), NAME

```

```

340   FORMAT (A, '[' I1, '; 4H', A8, 3X, A8, 2X, F8.2, 4X, F8.2,
*           4X, F8.2, 6X, F7.5, 4X, A3)
345   FORMAT (A, '[' I2, '; 4H', A8, 3X, A8, 2X, F8.2, 4X, F8.2,
*           4X, F8.2, 6X, F7.5, 4X, A3)

      KONT = KONT + 1
350 CONTINUE
      CALL ENCA
      KONT = 7
      CALL CUADRO
      WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
      WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
      WRITE (*, 366) ESC, ESC
366 FORMAT(A, '[5; 9H', ' NODO-ENVIO', ' NODO-RECEP', 4X, ' RESISTENCIA',
*         /, A, '[5; 49H', ' REACTANCIA Y-DERI / 2')
      DO 400 I = 1, NLines
      JX = NE(I)
      JY = NR(I)
      NOMB1 = DNOMB(JX)
      NOMB2 = DNOMB(JY)
      IF (KONT.LT.23) GO TO 370
      KONT = 7
      CALL ENCA
      CALL CUADRO
      WRITE (*, 366) ESC, ESC
      WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
      WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
370   IF (KONT.LT.10)
370   IF (KONT.LT.10) WRITE (*, 380) ESC, KONT, NOMB1, NOMB2, R(I), X(I), YPQ2E(I)
      IF (KONT.GE.10) WRITE (*, 385) ESC, KONT, NOMB1, NOMB2, R(I), X(I), YPQ2E(I)
380   FORMAT (A, '[' I1, '; 10H', A8, 4X, A8, 5X, F8.5, 6X, F8.5, 6X, F8.5)
385   FORMAT (A, '[' I2, '; 10H', A8, 4X, A8, 5X, F8.5, 6X, F8.5, 6X, F8.5)
      KONT = KONT + 1
400 CONTINUE
      IF (NTAPS.EQ.0) GO TO 455
      CALL ENCA
      KONT = 7
      CALL CUADRO
      WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
      WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
      WRITE (*, 406) ESC
406 FORMAT(A, '[5; 6H', ' NODO-CON-TAP', 5X, ' NODO-SIN-TAP', 8X, ' VALOR-TAP')
      DO 450 I = 1, NTAPS
      JX = NETR(I)
      JY = NRTR(I)
      NOMB1 = DNOMB(JX)
      NOMB2 = DNOMB(JY)
      IF (KONT.LT.23) GO TO 430
      KONT = 7
      CALL ENCA
      CALL CUADRO
      WRITE (*, 406) ESC
      WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
      WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2

```

```

430     IF (KONT.LT.10)
430     IF (KONT.LT.10) WRITE (*,440)ESC,KONT,NOMB1,NOMB2,TAP(I)
        IF (KONT.GE.10) WRITE (*,445)ESC,KONT,NOMB1,NOMB2,TAP(I)
440     FORMAT (A,[' I1, ';8H',A8,9X,A8,10X,F8.5)
445     FORMAT (A,[' I2, ';8H',A8,9X,A8,10X,F8.5)
        KONT = KONT + 1
450 CONTINUE
455 IF (NSHNTS.EQ.0) GO TO 600
        CALL ENCA
        KONT = 7
        CALL CUADRO
        WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
        WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
        WRITE (*,466) ESC
466 FORMAT(A,['5;15H', 'NODO-SHUNT           SUSCEPTANCIA' )
        DO 500 I = 1, NSHNTS
            JX = NSHUNT(I)
            NOMB1 = DNOMB(JX)
            IF (KONT.LT.23) GO TO 470
            KONT = 7
            CALL ENCA
            CALL CUADRO
            WRITE (*, 466) ESC
            WRITE (*, 44) ESC, ESC, WHITEFORE, LETR1
            WRITE (*, 45) ESC, ESC, WHITEFORE, LETR2
470     IF (KONT.LT.10)
470     IF (KONT.LT.10) WRITE (*,480)ESC,KONT,NOMB1,SHUNT(I)
        IF (KONT.GE.10) WRITE (*,485)ESC,KONT,NOMB1,SHUNT(I)
480     FORMAT (A,[' I1, ';16H',A8,13X,F8.5)
485     FORMAT (A,[' I2, ';16H',A8,13X,F8.5)
        KONT = KONT + 1
500 CONTINUE

```

Se imprime reporte de datos por impresora o archivo. Si ARCH2 = 'PRN', es por impresora. Si ARCH3 ≠ ' ', se imprime en archivo.

```

600 IF (PRINTR.EQ.0) GO TO 1000
        IF (ARCH2.EQ.'PRN') CALL CUADRO
        IF (ARCH2.EQ.'PRN') WRITE (*,607) ESC
        IF (ARCH2.EQ.'PRN') CALL ENCA
        ARCH21 = ARCH2
        ARCH31 = ARCH3
601 IF (ARCH31.EQ.' ') GO TO 602
        NAME = ARCH31
        ARCH31 = ' '
        GO TO 605
602 IF (ARCH21.EQ.' ') GO TO 1000
        NAME = ARCH21
        ARCH21 = ' '

```

Se abre archivo de impresión.

```

605 OPEN (UNIT = IWR, FILE = NAME , ERR = 810 )
607 FORMAT(A,['10;9H', 'ENCIENDA Y COLOQUE EN LINEA LA IMPRESORA...')

```

```

WRITE (IWR, 43) LETR1
WRITE (IWR, 43) LETR2
WRITE (IWR, 615)
615 FORMAT(/10X, ' NODO ', 2X, 'P-CARGA (MW)', 2X, 'Q-CARGA (MVAR)')
DO 630 I = 1, NNODOS
    WRITE (IWR, 620) DNOMB(I), PC(I), QC(I)
620     FORMAT (10X, A8, 3X, F8.2, 4X, F8.2)
630 CONTINUE
    WRITE (IWR, 640)
640 FORMAT (/3X,
* 'NODO-GEN NODO-ALTA P-GEN(MW) QG-MAX(MVAR) QG-MIN(MVAR)',
* ' V-GEN (PU) TIPO')
DO 670 I = 1, NNGEN
    JX = NPV(I)
    JY = NAU(I)
    JY = ABS(JY)
    NOMB1 = DNOMB(JX)
    NOMB2 = DNOMB(JY)
    WRITE (IWR, 650) NOMB1, NOMB2, PG(I), QGMAX(I),
*                   QGMIN(I), VESP(I), TIPGEN(I)
650     FORMAT (3X, A8, 2X, A8, 3X, F8.2, 2(4X, F9.2), 4X, F7.5, 5X, I3)
670 CONTINUE
    WRITE (IWR, 675)
675 FORMAT(/10X, 'NODO-ENVIO NODO-RECEP RESISTENCIA REACTANCIA',
*           ' Y-DERI / 2')
DO 700 I = 1, NLLINES
    JX = NE(I)
    JY = NR(I)
    NOMB1 = DNOMB(JX)
    NOMB2 = DNOMB(JY)
    WRITE (IWR, 685) NOMB1, NOMB2, R(I), X(I), YPQ2E(I)
685     FORMAT (12X, A8, 4X, A8, 5X, F8.5, 5X, F8.5, 5X, F8.5)
700 CONTINUE
    IF (NTAPS.EQ.0) GO TO 750
    WRITE (IWR, 710)
710 FORMAT (/10X, 'NODO-CON-TAP NODO-SIN-TAP VALOR-TAP')
DO 740 I = 1, NTAPS
    JX = NETR(I)
    JY = NRTR(I)
    NOMB1 = DNOMB(JX)
    NOMB2 = DNOMB(JY)
    WRITE (IWR, 735) NOMB1, NOMB2, TAP(I)
735     FORMAT (12X, A8, 9X, A8, 10X, F8.5)
740 CONTINUE
750 IF (NSHNTS.EQ.0) GO TO 3000
    WRITE (IWR, 760)
760 FORMAT (/10X, 'NODO-SHUNT SUSCEPTANCIA-DERIVACION')
DO 790 I = 1, NSHNTS
    JX = NSHUNT(I)
    NOMB1 = DNOMB(JX)
    WRITE (IWR, 780) NOMB1, SHUNT(I)
780     FORMAT (11X, A8, 15X, F8.5)
790 CONTINUE

```

```

GO TO 900
810 IF (NAME.NE.ARCH2) GO TO 900
    CALL CUADRO
    WRITE (*,815) ESC
815 FORMAT (A,'[10;10H','IMPRESORA FUERA DE LINEA... REVISELA!!!')
    CALL ENCA

```

Se cierra archivo de impresión de reporte de datos.

```

900 CLOSE (UNIT = IWR)
    GO TO 601

```

Se convierten MW y MVars a valores en por unidad (pu).

```

1000 DO 1020 I = 1, NNODOS
        PG(I) = PG(I) / SBASE
        QG(I) = QG(I) / SBASE
        PC(I) = PC(I) / SBASE
        QC(I) = QC(I) / SBASE
1020 CONTINUE
    DO 1040 I = 1, NNGEN
        QGMAX(I) = QGMAX(I) / SBASE
        QGMIN(I) = QGMIN(I) / SBASE
1040 CONTINUE
3000 CONTINUE

```

Fin del subprograma LEDATO.

```

RETURN
END

```

c) Subprograma PANTAL

La actividad de este subprograma es generar la pantalla de presentación del simulador de flujos de potencia.

La instrucción para acceder este subprograma es:

```
CALL PANTAL
```

Este subprograma no involucra argumentos al ser llamado. El código correspondiente, así como una explicación del mismo se muestra a continuación.

SUBROUTINE PANTAL

```

C*****
C* SUBPROGRAMA REALIZADO POR: NAHUM ROMAN VARGAS. *
C* MODIFICADO POR: JOSE HORACIO TOVAR HERNANDEZ. *
C*****

```

Declaración de variables.

```

CHARACTER ESC*1, HOME*3, STAY*3
PARAMETER (HOME=' [2J', STAY=' [s')
CHARACTER WHITBACK*4, CYANBACK*4, RESET*4
PARAMETER (WHITBACK=' [47m', CYANBACK=' [46m', RESET=' [00m')
CHARACTER BLUEBACK*4, YELLFORE*7, BLACBACK*4
PARAMETER (BLUEBACK=' [44m', YELLFORE=' [33;01m', BLACBACK=' [40m')
CHARACTER BLUEFORE*4, CYANFORE*4, BLACFORE*4
PARAMETER (BLUEFORE=' [34m', CYANFORE=' [36m', BLACFORE=' [30m')

```

```

ESC=CHAR(27)
1 FORMAT(30A)
2 FORMAT(3A, '[', I2.2, ';', I2.2, 'H', 31(' '), A, A)
3 FORMAT(3A, '[', I2.2, ';', I2.2, 'H', 2A)
4 FORMAT(4A, 32(' '), 2A)

```

Se pone el modo gráfico normal en pantalla (caracteres blancos y fondo negro).

```
WRITE(6,1) ESC, RESET
```

Se duplica el tamaño de los caracteres.

```
WRITE(6,1) ESC, '[=1h'
```

Pinta fondo en color 'cyan'. Se posiciona el cursor en primer renglón y primera columna.

```
WRITE(6,1) ESC, CYANBACK, ESC, HOME
```

Se pinta un cuadro en color azul y efecto de sombra en color negro.

```

DO 5 I = 1, 13
  WRITE(6,2) ESC, BLUEBACK, ESC, I+6, 5, ESC, STAY
  IF(I.EQ.1) GO TO 5
  WRITE(6,3) ESC, BLACFORE, ESC, I+6, 37, ESC, '[u'
5 CONTINUE
WRITE(6,4) ESC, BLACFORE, ESC, '[20;6H', ESC, '[u'
WRITE(6,1) ESC, BLUEBACK, ESC, '[u'

```

Se pintan letrero de presentación letra por letra y con un retardo de tiempo entre cada una de ellas.

```

WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C'
N1 = 100
N2 = 200
  CALL CICLO (N1, N2)
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C '
CALL CICLO (N1, N2)

```



```
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E P '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E P '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E P A '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E P A '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, YELLFORE, ESC, '[09;15H C E P A L '  
WRITE(6,1) ESC, '[11;05H P '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PR '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PRO '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROG '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRA '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAM '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA D '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE F '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FL '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLU '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJ '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJO '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS D '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE P '  
CALL CICLO (N1, N2)  
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE PO '  
CALL CICLO (N1, N2)
```

```

WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POT'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POTE'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POTEN'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POTENC'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POTENCI'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[11;05H PROGRAMA DE FLUJOS DE POTENCIA'
  N1 = 300
  N2 = 300
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;07H J'
  N1 = 100
  N2 = 200
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;07H JH'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;07H JHT'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;07H JHTH'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;28H1'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;28H19'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;28H199'
CALL CICLO (N1, N2)
WRITE(6,1) ESC, '[17;28H1992', ESC, STAY
DO 7 I = 1, 1400
DO 7 J = 1, 1000
7 CONTINUE

```

Se reinicializa el modo normal de la pantalla.

```
WRITE(6,1) ESC, RESET, ESC, '[=3h', ESC, '[u'
```

Se termina el subprograma PANTAL.

```
RETURN
END
```

5. Descripción de Subprogramas para Actividades de Salida

En estas actividades se crean los distintos reportes de resultados que el simulador ofrece: flujos, pérdidas, parámetros, condiciones iniciales para el programa de estabilidad y resúmenes. Además, ofrece la oportunidad de modificar el archivo de datos del sistema eléctrico de potencia, definido al pasar por el subprograma PORTAD. Este archivo, una vez modificado permite crear otro con cualquier nombre y direccionamiento, para ser utilizado en simulaciones posteriores, o efectuar otra simulación durante la misma sesión.

Tales actividades se encuentran desarrolladas en los subprogramas FINALE, DATREP, RESREP Y TOPCAM, los cuales se describen a continuación.

a) Subprograma FINALE

Las actividades de este subprograma son:

Creación de reportes de flujos y/o pérdidas, ya sea completos, por isla eléctrica o por nodo. Tales reportes pueden enviarse a través de pantalla o un archivo tipo ASCII que pueda ser consultado posteriormente.

La forma de acceder este subprograma es mediante la instrucción:

```
CALL FINALE (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, YPQ2E, YPQ2R, VRE,
*          VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT, SHUNT, DNOMB,
*          NISLAS, INISLA, ISLA, NAU, NPV, NNGEN, YM, ABORTA, LETR1,
*          LETR2, QGMIN, QGMAX, KFUERA, NODFUE, ISLACK, GPQ2E, GPQ2R)
```

Una descripción de las variables involucradas en el argumento del subprograma, así como del bloque común CIRCUI se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
PG(NNODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema eléctrico de potencia.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema eléctrico de potencia.
PC(NNODOS)	E	REAL	Potencia real de carga en cada nodo del sistema eléctrico de potencia.
QC(NNODOS)	E	REAL	Potencia reactiva de carga en cada nodo del sistema eléctrico de potencia.
TIPO(NNODOS)	E	ENTERA	Tipo de cada nodo del sistema eléctrico.
NLT(NNODOS)	E	ENTERA	Indica si el nodo asociado a la posición contiene un compensador en derivación: si $NLT(i) = 0$, el nodo i no contiene compensación en derivación; si $NLT(i) = 1$, el nodo i si la contiene.
R1(NLINES)	E	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	E	REAL	Susceptancia inductiva serie del circuito π de cada elemento de transmisión.
YPQ2E(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de envío en el circuito π de cada elemento de transmisión.
YPQ2R(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de recepción en el circuito π de cada elemento de transmisión.
VRE(NNODOS)	E	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	E	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
PCAL(NNODOS)	E/S	REAL	Pot. real neta inyectada en cada nodo del sistema.
QCAL(NNODOS)	E/S	REAL	Potencia reactiva neta inyectada en cada nodo del sistema eléctrico.
SBASE	E	REAL	Potencia base del sistema eléctrico en MVA.
NSHNTS	E	ENTERA	Total de compensadores en derivación en el sistema eléctrico de potencia.
NSHUNT(NSHNTS)	E	ENTERA	Indica el nodo en donde se conecta un compensador en derivación.
SHUNT(NSHNTS)	E	REAL	Valor de la susceptancia del compensador en derivación conectado al nodo indicado en NSHUNT(.).
DNOMB(NNODOS)	E	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema eléctrico.
INISLA(NISLAS)	E	ENTERA	Indica el número de nodo en donde se inicia la información topológica de la isla indicada por la posición de INISLA(.).
ISLA(NISLAS)	E	ENTERA	Contiene los nodos del sistema, en forma de listas encadenadas para cada isla.
NNGEN	E	ENTERA	Total de nodos generadores en el sistema eléctrico.
NAU(NNGEN)	E	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
NPV(NNGEN)	E	ENTERA	Contiene en cada posición el número correspondiente a un nodo de generación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
YM(NNGEN)	E	REAL	Susceptancia equivalente de compensadores estáticos de potencia reactiva con límites violados.
ABORTA	E/S	ENTERA	Indica si al salir de este subprograma se ejecuta la siguiente instrucción del programa principal (ABORTA = 0) ó se termina la sesión de simulación (ABORTA = 1).
LETR1	E	CHARACTER	Primer renglón descriptivo del archivo de datos del sistema eléctrico.
LETR2	E	CHARACTER	Segundo renglón descriptivo del archivo de datos del sistema eléctrico.
QGMIN(NNGEN)	E	REAL	Límite mínimo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).
QGMAX(NNGEN)	E	REAL	Límite máximo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).
KFUERA	E	ENTERA	Total de nodos que han sido desconectados del sistema en sesiones previas en el subprograma TOPCAM.
NODFUE(KFUERA)	E	ENTERA	Indica el número de nodo desconectado del sistema en sesiones previas del subprograma TOPCAM.
ISLACK(NISLAS)	E	ENTERA	Indica el número de nodo asignado como compensador en cada isla eléctrica.
GPQ2E(NLINES)	E	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de envío.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
GPQ2R(NLINES)	E	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de recepción.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica donde se inicia la información referente a la posición de circuitos π , la cual se encuentra en IPOSPR(), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	NE(i) indica el nodo de envío correspondiente al circuito π i-ésimo.
NR(NLINES)	E	ENTERA	NR(i) indica el nodo de recepción correspondiente al circuito π i-ésimo.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k) indica donde se encuentra un circuito π relacionado con el nodo i, donde k está definida por IFINPR(.).
NTAPS	E	ENTERA	Total de transformadores con tap variable.
NETR(NTAPS)	E	ENTERA	Identificador del nodo al lado del tap de transformadores con tap variable.
NRTR(NTAPS)	E	ENTERA	Identificador del nodo al lado contrario del tap de transformadores con tap variable.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
TAP(NTAPS)	E	REAL	Valor del tap en cada uno de los transformadores con tap variable.
VALNUM(15)	E/S	REAL	Almacena números reales que corresponden a variables involucradas en un menú.
DEPEND(15)	E/S	ENTERA	Indica dependencia entre opciones de un menú.
LISTOP(15)	E/S	CHARACTER	Contiene las cadenas de caracteres alfanuméricos describiendo cada opción del menú.
ENCABE	E/S	CHARACTER	Cadena de caracteres alfanuméricos para identificar cada menú.
ALTERN	E/S	CHARACTER	Cadena de caracteres alfanuméricos para formular la pregunta de selección de opción.
TIPVAL(15)	E/S	CHARACTER	Cadena de tres caracteres alfanuméricos indicando el tipo de variable asociada con cada opción.
VALCHA(15)	E/S	CHARACTER	Contiene variables de tipo alfanumérico asociadas con alguna opción del menú.

Las demás variables utilizadas en este subprograma se describen conforme se explica cada parte de código del mismo, el cual se presenta a continuación.

```

C*****
C  SUBPROGRAMA FINALE.
C  REALIZADO POR: JOSE HORACIO TOVAR HERNANDEZ.
C                                     OCTUBRE DE 1992.
C*****

```

Inicio del Subprograma FINALE.

```

C
C  CALCULOS FINALES DE FLUJOS.
C

```



```

SUBROUTINE FINALE (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, YPQ2E, YPQ2R,
*
*           VRE, VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT, SHUNT,
*
*           DNOMB, NISLAS, INISLA, ISLA, NAU, NPV, NNGEN, YM,
*
*           ABORTA, LETR1, LETR2, QGMIN, QGMAX, KFUERA, NODFUE,
*
*           ISLACK, GPQ2E, GPQ2R)

```

Declaración de bloques comunes.

```

COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
COMMON /CIRCU1/ NLINE, INICPR, IFINPR, NE, NR, IPOSPR

```

Definición y dimensionamiento de variables.

```

REAL*8      KG, VALNUM(15)

C
INTEGER     NE(1000), NR(1000), NETR(300), NRTR(300), NLT(550), OPCION,
*           INICPR(550), NAU(150), INISLA(10), ISLA(550), IFINPR(2000),
*           IPOSPR(2000), NSHUNT(150), NPV(150), TIPO(550), ABORTA,
*           DEPEND(15), NODFUE(15), ISLACK(10)

REAL        PG(550), QG(550), PC(550), QC(550), PCAL(550), YPQ2E(1000),
*           R1(1000), X1(1000), YPQ2R(1000), TAP(300), TPG(10), TQG(10),
*           VRE(550), QCAL(550), TQC(10), PLS(10), QLS(10), TPC(10),
*           SHUNT(150), QGMIN(150), QGMAX(150), VIM(550), YM(150),
*           GPQ2E(1000), GPQ2R(1000), MP, MQ

C
CHARACTER   LISTOP(15)*50, ENCABE*50, TIPVAL(15)*3, ARCH*40
CHARACTER   ESC*1, DISPOS*2, ES*1, LETR1*70, LETR2*70
CHARACTER   WHITEFORE*4, YELFORE*4, ALTERN*50
PARAMETER   (YELFORE='[33m', WHITEFORE = '[37m')
CHARACTER   NOMB*8, DNOMB(550)*8, NOMB1*8, NAME*12, VALCHA(15)*12

```

Se inicializa la variable ESC, la cual sirve para crear secuencias de 'escape' en pantalla. ABORTA se define para NO salir inmediatamente del programa principal. Se define a LTERM como canal de salida por pantalla. El subprograma CUADRO limpia y pinta un margen en pantalla

```

ESC = CHAR(27)
ABORTA = 0
LTERM = 6
CALL CUADRO

```

Se selecciona el tipo de reporte deseado: Si se teclea PL, el reporte será de pérdidas (KO = 0). Si se teclea FL o "entrada", el reporte será de flujos (KO = 1).

```

25 NOMB = 'FL'
WRITE (*, 30) ESC, ESC, YELFORE
WRITE (*, 35) ESC, ESC, WHITEFORE, NOMB, ESC
30 FORMAT (A, '[10;14H', A, A4,
*         '¿ REPORTE DE PERDIDAS O DE FLUJOS [PL, FL] ? ')
35 FORMAT (A, '[10;69H', A, A4, A2, A, '[10;69H', $)
READ(LTERM, 200) NOMB1

```

```

KO = - 1
IF (NOMB1.EQ.'FL'.OR.NOMB1.EQ.'f1'.OR.NOMB1.EQ.'fL'.OR.
*      NOMB1.EQ.'F1'.OR.NOMB1.EQ.' ') KO = 1
IF (NOMB1.EQ.'PL'.OR.NOMB1.EQ.'pl'.OR.NOMB1.EQ.'pL'.OR.
*      NOMB1.EQ.'P1')          KO = 0
IF (KO.LT.0) GO TO 25

```

IWR se define como canal de salida. Se limpia pantalla con CLEAR y se inicializa un menú con INIMEN.

```

50 IWR = LTERM
CALL CLEAR
CALL INIMEN

```

Se especifican las características del menú para reportes.

```

ENCABE = 'MENU PARA REPORTES DE FLUJOS : '
IF(KO.EQ.0) ENCABE = 'MENU PARA REPORTES DE PERDIDAS: '
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' REGRESAR A MENU DE REPORTES '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' REPORTE COMPLETO '
LISTOP(4) = ' REPORTE POR NODO '
IF (NISLAS.EQ.1) GO TO 51
LISTOP(5) = ' REPORTE POR ISLA '

```

Se define a la opción 1 como la de 'default'. Se llama al subprograma MENU.

```

51 OPCION = 1
CALL MENU (OPCION)
IF (OPCION.EQ.3)          GO TO 170
IF (OPCION.EQ.4.OR.OPCION.EQ.5) GO TO 52
IF (OPCION.EQ.1)          GO TO 3000
IF (OPCION.NE.2)          GO TO 50
ABORTA = 1
RETURN

```

Se inician las variables KISLA (número de isla), NOMB1 para leer nombre de un nodo, IFLAG para regresar a la parte de reporte por nodo o isla.

```

52 KISLA = 0
NOMB1 = ' '
IFLAG = 0
CALL CUADRO

```

Se escribe en pantalla el título del tipo de reporte seleccionado.

```

IF (KO.EQ.1.AND.OPCION.EQ.3) WRITE (LTERM,55) ESC, ESC
55 FORMAT (A,'[3;25H','REPORTE DE FLUJOS COMPLETO',/,
*      A,'[4;25H','-----')
IF (KO.EQ.0.AND.OPCION.EQ.3) WRITE (LTERM,58) ESC, ESC
58 FORMAT (A,'[3;25H','REPORTE DE PERDIDAS EN ELEMENTOS',/,
*      A,'[4;25H','-----')
IF (KO.EQ.1.AND.OPCION.EQ.4) WRITE (LTERM,60) ESC, ESC

```

```

60 FORMAT (A, '[3;25H', 'REPORTE DE FLUJOS POR NODO', '/',
*      A, '[4;25H', '----- -- ----- ----')
      IF (KO.EQ.0.AND.OPCION.EQ.4) WRITE (LTERM,62) ESC, ESC
62 FORMAT (A, '[3;25H', 'REPORTE DE PERDIDAS POR NODO', '/',
*      A, '[4;25H', '----- -- ----- ----')
      IF (OPCION.EQ.5) WRITE (LTERM,64) ESC, ESC
64 FORMAT (A, '[3;25H', 'REPORTE DE FLUJOS POR ISLA', '/',
*      A, '[4;25H', '----- -- ----- ----')
      IF (OPCION.EQ.5) WRITE (LTERM,65) ESC, ESC
65 FORMAT (A, '[3;25H', 'REPORTE DE PERDIDAS POR ISLA', '/',
*      A, '[4;25H', '----- -- ----- ----')

```

Se manda a especificar si el reporte es por pantalla o impresora.

GO TO 175

Después de especificar el canal de salida, se pide nombre de nodo o número de isla, según se haya especificado en el menú de reportes.

```

66 WRITE (LTERM,69) ESC
      IF (OPCION.EQ.4) WRITE (LTERM,70) ESC, ESC
      IF (OPCION.EQ.5) WRITE (LTERM,72) ESC, NISLAS, ESC
69 FORMAT (A, '[24;4H', '
*      '
70 FORMAT (A, '[24;6H',
*      '¿ QUE NODO [maximo 8 caracteres, return para salir] ?',
*      '
*      '
A, '[24;61H', '$)
72 FORMAT (A, '[24;6H', '¿ CUAL ISLA [ num. entero entre 1 y ', I2,
*      ' ' return ' para salir ] ? ', '/', A, '[24;73H', '$)
      IF (OPCION.EQ.4) READ (LTERM,80) NOMB1
      IF (OPCION.EQ.5) READ (LTERM,81) KISLA
      WRITE (LTERM,78) ESC
78 FORMAT (A, '[16;10H', '
80 FORMAT(A8)
81 FORMAT(I2)

```

Se verifica si lo que se leyó corresponde a un nodo o isla, o si se desea salir de tales opciones (NOMB1=' ', KISLA=0, para salir).

```

      IF (OPCION.EQ.4.AND.NOMB1.NE.' ') GO TO 85
      IF (OPCION.EQ.5.AND.KISLA.NE.0) GO TO 83
      IF (NOMB.EQ.'IM') CLOSE (UNIT = 8)

```

Se sale de la opción ya seleccionada y se regresa al menú de reportes.

GO TO 50

Se verifica que la información leída sea correcta (KISLA debe corresponder a una isla existente en el sistema eléctrico en estudio).

```

83 IF (KISLA.LE.NISLAS) GO TO 340
      WRITE (*,84) ESC, KISLA
84 FORMAT (A, '[19;10H', ' ** LA ISLA ', I2, ' NO EXISTE ')
      GO TO 66

```

Se verifica que NOMB1 sea un nombre de nodo correcto.

```
85 DO 90 J = 1, NNODOS
    IF(NOMB1.NE.DNOMB(J))GO TO 90
```

Se verifica que el nodo J, correspondiente al nombre dado en NOMB1, no sea un nodo dado de baja en el subprograma TOPCAM.

```
    DO 86 L = 1, KFUERA
        IF (NODFUE(L).NE.J) GO TO 86
        WRITE (*,101) ESC, NOMB1
        GO TO 103
86 CONTINUE
```

Se manda el control para verificar que el nodo J no ha sido reducido, por ser un compensador estático de VARS con límites violados.

```
        GO TO 150
90 CONTINUE
```

Se escribe mensaje de que el nodo propuesto no existe en el sistema.

```
        WRITE (LTERM,100) ESC, NOMB1
100 FORMAT(A,'[19;10H',' ** EL NODO ',A8,' NO EXISTE ')
101 FORMAT(A,'[19;10H',' ** EL NODO ',A8,' FUE REMOVIDO DEL SISTEMA')
102 FORMAT(A,'[19;10H',' ** EL NODO ',A8,' FUE REDUCIDO ')
103 IFLAG = 1
```

Se regresa el control a leer otro nombre de nodo u otro número de isla

```
GO TO 66
```

Se revisa si el nodo no se redujo por ser un CEV con límites violados.

```
150 DO 154 I1 = 1, NNODOS
    IF (ISLA(I1).EQ.J) GO TO 160
    IF (ABS(ISLA(I1)).EQ.J) GO TO 155
154 CONTINUE
```

Se escribe mensaje de que el nodo fué reducido.

```
155 WRITE(6,102) ESC, DNOMB(I1)
```

Se regresa el control a leer otro nombre de nodo u otro número de isla

```
GO TO 66
160 CONTINUE
```

Se envía mensaje a pantalla de grabación de información del nodo leído.

```
IF (NOMB.EQ.'IM') WRITE (LTERM,161) ESC, NOMB1
161 FORMAT(A,'[19;10H',' SE GRABO INFORMACION DEL NODO ',A8)
```

Secuencia de borrado de pantalla aplicando un efecto de atraso.

```
IFLAG = 0
ILE = 9
```

```

ISIZE = 25
DO 167 KJ = 1, LIMPIA
  DO 166 I1 = 5, 18
    IF (I1.LT.10) WRITE (LTERM,163) ESC, I1
    IF (I1.GE.10) WRITE (LTERM,164) ESC, I1
163   FORMAT (A,'[',I1,',';4H',' ,',
*       ',')
164   FORMAT (A,'[',I2,',';4H',' ,',
*       ',')
166   CONTINUE
167   CONTINUE
      DO      I1 = 19, 22
        WRITE (LTERM,164) ESC, I1
      END DO
      WRITE (LTERM,168) ESC
168   FORMAT (A,'[7;4H',' ,',
*       ',')
      WRITE (LTERM,169) ESC, ESC
169   FORMAT (A,'[3;4H',' ,',
*       ',')
*     /,A,'[4;4H',' ,',
*     ',')

```

Se inicializa contador de renglones para hacer posible el paginado por pantalla.

```
ILE = 1
```

Se transfiere control para iniciar secuencia de impresión.

```
GO TO 340
```

Se inicia secuencia de impresión de los distintos reportes.

```

170 CONTINUE
    IFLAG = 1
    CALL CUADRO
    IF (KO.EQ.1) WRITE (LTERM,172) ESC, ESC
172 FORMAT (A,'[4;25H',' REPORTE DE FLUJOS COMPLETO ',/,
*         A,'[5;25H',' ----- -- ----- ')
    IF (KO.EQ.0) WRITE (LTERM,174) ESC, ESC
174 FORMAT (A,'[4;25H',' REPORTE DE PERDIDAS COMPLETO ',/,
*         A,'[5;25H',' ----- -- ----- ')

```

Se define el canal de salida: pantalla o impresora.

```

175 NOMB = 'PA'
    WRITE (LTERM, 180) ESC, ESC, YELFORE
    WRITE (LTERM, 190) ESC, ESC, WHITEFORE, NOMB, ESC
180 FORMAT (A,'[10;14H', A, A4,
*         '( REPORTE EN IMPRESORA O PANTALLA [IM, PA] ? ')
190 FORMAT (A,'[10;69H',A,A4,A2,A,'[10;69H',A,$)
    READ(LTERM,200) NOMB
200 FORMAT(A2)
    WRITE (LTERM, 202) ESC, ESC, YELFORE

```

```

WRITE (LTERM, 205) ESC, ESC, WHITEFORE, ESC
202 FORMAT (A, '[10;14H', A, A4,
*
205 FORMAT (A, '[10;69H', A, A4, ' ', A, '[10;69H', $)
IF(NOMB.EQ. 'IM'.OR. NOMB.EQ. 'im')GO TO 240
IF(NOMB.EQ. 'PA'.OR. NOMB.EQ. 'pa'.OR. NOMB.EQ. ' ') GO TO 320
GO TO 175

```

Se indica que el dispositivo es la impresora.

```

240 DISPOS = 'IM'
LIMPIA = 7

```

Se define el archivo de salida a impresora de 'default'.

```

ARCH = 'FLUJOS.SAL'

```

Se pregunta por el nombre del archivo de salida a impresora.

```

WRITE (LTERM, 250) ESC, ESC, YELFORE
WRITE (LTERM, 260) ESC, ESC, WHITEFORE, ARCH, ESC
250 FORMAT (A, '[15;3H', A, A4,
* '( NOMBRE DEL ARCHIVO PARA REPORTE ? ' )
260 FORMAT (A, '[15;40H', A, A4, A12, A, '[15;40H', $)
READ(LTERM, 270) ALTERN
270 FORMAT(A40)

```

Si se responde a la pregunta por nombre de archivo con entrada, el nombre es FLUJOS.SAL.

```

IF (ALTERN.EQ. ' ') GO TO 280
ARCH = ALTERN

```

Se define el canal de escritura para el archivo de salida y se abre el archivo que contendrá el reporte.

```

280 IWR = 8
OPEN (UNIT = 8, FILE = ARCH )

```

Se imprime en el archivo de reporte la descripción del archivo de datos

```

WRITE(IWR,300)LETR1,LETR2
300 FORMAT(/1X,A70,/ ,1X,A70)
NOMB = 'IM'

```

Se regresa el control para preguntar por otro nodo u otra isla, en caso de que las instrucciones anteriores hayan sido aplicadas sin provenir de la opción 3 del menú de reportes.

```

IF (IFLAG.EQ.0) GO TO 66

```

Se transfiere control para iniciar reporte para impresora.

```

GO TO 340

```

Se inician variables auxiliares para impresión de reporte por pantalla.

```

320 DISPOS = 'PA'
    LIMPIA = 1
    ISIZE = 21
    IWR = LTERM

```

Se regresa el control para preguntar por otro nodo u otra isla, en caso de que las instrucciones anteriores hayan sido aplicadas sin provenir de la opción 3 del menú de reportes.

```
IF (IFLAG.EQ.0) GO TO 66
```

Se inicializa contador para reporte por pantalla.

```

340 IF (NISLAS.EQ.1) ILE = 1
    IF (NISLAS.GT.1) ILE = 4

```

Se preparan variables auxiliares para establecer límites en el reporte, esto es, por nodo, por isla o completo.

```

371 INIC1 = 1
    IFIN1 = NISLAS
    IF (IFLAG.NE.0) GO TO 372
    INICIO = 1
    IFIN1 = 1
    IFIN = 1
    IF (OPCION.NE.5) GO TO 372
    INIC1 = KISLA
    IFIN1 = KISLA

```

Se inicia un ciclo para imprimir reporte, en base a los límites dados.

```
372 DO 720 KISLA = INIC1, IFIN1
```

PL = pérdidas de potencia real, QL = pérdidas de potencia reactiva,
TOTPG = total de potencia real generada,
TOTQG = total de potencia reactiva generada,
TOTPC = total de potencia real de carga,
TOTQC = total de potencia reactiva de carga.

```

PL = 0.0
QL = 0.0
TOTPG = 0.0
TOTQG = 0.0
TOTPC = 0.0
TOTQC = 0.0

```

Se transfiere control a reporte por nodo.

```
IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 384
```

Se imprime la isla correspondiente al reporte por impresora.

```
IF (DISPOS.EQ.'IM'.AND.NISLAS.GT.1) WRITE(IWR,380) KISLA
```

Se inicializa pantalla para reporte por este dispositivo.

```
IF (DISPOS.EQ.'PA') CALL CUADRO
```

Se imprime en pantalla número de la isla actual.

```

      IF (DISPOS.EQ.'PA'.AND.NISLAS.GT.1)
*           WRITE (IWR,382) ESC, ESC, KISLA, ESC
380  FORMAT(//20X,'*****',
*           /20X,'RESULTADOS DE LA ISLA No. ',I2,
*           /20X,'*****',//)
382  FORMAT(A,'[2;20H','*****',/,
*           A,'[3;20H','RESULTADOS DE LA ISLA No. ',I2,/,
*           A,'[4;20H','*****')

```

IRENGL = número de renglón inicial para reporte paginado por pantalla.
IANCHO = número máximo de renglones en una página de reporte.

```

IRENGL = 2
IANCHO = 23

```

Así se define IRENGL para imprimir antes el letrero del formato 382.

```

      IF (NISLAS.GT.1) IRENGL = 5

```

Se especifican límites para el reporte por isla.

```

      INICIO = INISLA(KISLA)
      IF (KISLA.LT.NISLAS) IFIN = INISLA(KISLA + 1) - 1
      IF (KISLA.EQ.NISLAS) IFIN = NNODOS
384  DO 600 JS = INICIO, IFIN
      I = ISLA(JS)

```

Se verifica que el nodo I sea un nodo no reducido en el sistema.

```

      IF (I.LT.0) GO TO 600
      IF (TIPO(I).LT.0) GO TO 600
      DO I1 = 1, KFUERA
          IF (I.EQ.NODFUE(I1)) GO TO 600
      END DO

```

El nodo I es actual por lo que es factible imprimir su información.

```

      IF (IFLAG.EQ.0.AND.OPCION.EQ.4) I = J

```

El nodo J es el nodo seleccionado para el reporte por nodo.

```

      IF (I.LT.0) GO TO 590

```

Se inicializan desviaciones de potencia nodal y se suman potencias de generación y carga.

```

      MISMP = 0.0
      MISMQ = 0.0
      TOTPG = TOTPG + PG(I)
      IF(QG(I).GT.0.0)TOTQG = TOTQG + QG(I)
      IF(QG(I).LT.0.0)TOTQC = TOTQC - QG(I)
      TOTPC = TOTPC + PC(I)
      IF(QC(I).LT.0.0) TOTQG = TOTQG - QC(I)
      IF(QC(I).GT.0.0) TOTQC = TOTQC + QC(I)

```


Se calculan magnitud y ángulo de fase del voltaje complejo nodal.

```
V1 = VRE(I)
V2 = VIM(I)
VI = V1 * V1 + V2 * V2
VT = SQRT(VI)
IF (V1.NE.0.0) ANG = ATAN (V2 / V1)
IF (V2.EQ.0.0) ANG = 0.0
IF (V1.LT.0.0.AND.V2.LT.0.0) ANG=ANG - 3.141592654
IF (V1.GT.0.0.AND.V2.GT.0.0) ANG=ANG - (3.141592654*2.0)
IF (V1.LT.0.0.AND.V2.GT.0.0) ANG=ANG - 3.141592654
IF (V1.GT.0.0.AND.V2.LT.0.0) ANG=ANG
ANG = ANG * 57.29577951
```

Se expresan potencias real y reactiva en términos de MW y MVar.

```
A = PG(I) * SBASE
B = QG(I) * SBASE
C = 0.0
C = SQRT( A * A + B * B )
```

Se inicia el cálculo de desviaciones. Se contabiliza un renglón para imprimir potencias de generación.

```
MISMP = MISMP - A
MISMQ = MISMQ - B
IF (C.NE.0.0) ILE = ILE + 1
```

Se expresan potencias real y reactiva en términos de MW y MVar.

```
D = PC(I) * SBASE
E = QC(I) * SBASE
```

Se continúa el cálculo de desviaciones de potencia nodal.

```
MISMP = MISMP + D
MISMQ = MISMQ + E
F = 0.0
IF(D.EQ.0.0.AND.E.EQ.0.0) GO TO 435
```

Si la potencia de carga no es cero, se contabiliza un renglón para imprimir potencias de carga.

```
F = SQRT( D*D + E*E )
ILE = ILE + 1
```

Se verifica la existencia de compensación en derivación (NLT(I) ≠ 0).

```
435      G = 0.0
         IF(NLT(I).EQ.0) GO TO 440
         QM = 0.0
         IS = 0
436      IS = IS + 1
         IF(IS.GT.NSHNTS)GO TO 737
         IF(NSHUNT(IS).NE.1) GO TO 436
```

Se calcula la aportación de MVars del compensador en derivación.

```

QM = - VI * SHUNT(IS) * SBASE
MISMOQ = MISMOQ + QM
PM = 0.0
G = ABS( QM )
ILE = ILE + 1
IF(QM.LT.0.0) TOTQG = TOTQG - QM / SBASE
IF(QM.GT.0.0) TOTQC = TOTQC + QM / SBASE

```

Se contabilizan los renglones necesarios para imprimir toda la información referente al nodo I.

```

440      KC = INICPR(I)
          KC3 = 1
445      KC = IFINPR(KC)
          IF (KC.EQ.0) GO TO 447
          KC3 = KC3 + 1
          GO TO 445

```

Se verifica que tal información no rebase los 21 renglones disponibles.

```

447      IF ( (KC3+4) .LE. 21) GO TO 450
          CALL CLEAR
          ILE = 1
          GO TO 460

```

Se revisa si existen inyecciones cero para no imprimirlas.

```

450      KC3 = KC3 + ILE
          IF (C.NE.0.0) ILE = ILE - 1
          IF (F.NE.0.0) ILE = ILE - 1
          IF (G.NE.0.0) ILE = ILE - 1
          ILE = ILE + 1

```

Si se excede el máximo de renglones, se inicializa la pantalla.

```

          IF (KC3.LT. IANCHO.OR. DISPOS.EQ. 'IM') GO TO 470
          ILE = IRENGL
          CALL ENCA
          CALL CUADRO

```

Se imprime la información nodal en pantalla.

```

460      IF (OPCION.EQ. 3.AND. NISLAS.GT. 1)WRITE(*, 382)ESC, ESC, KISLA, ESC
          IF (OPCION.EQ. 5.AND. NISLAS.GT. 1)WRITE(*, 382)ESC, ESC, KISLA, ESC
470      IF (ILE.LE. 9.AND. DISPOS.EQ. 'PA'.AND. KO.EQ. 1)
          *      WRITE(IWR, 635) ESC, ILE, I, DNOMB(I), ESC, ILE, VT, ANG, I
          IF (ILE.GT. 9.AND. DISPOS.EQ. 'PA'.AND. KO.EQ. 1)
          *      WRITE(IWR, 637) ESC, ILE, I, DNOMB(I), ESC, ILE, VT, ANG, I
          IF (ILE.LE. 9.AND. DISPOS.EQ. 'PA'.AND. KO.EQ. 0)
          *      WRITE(IWR, 631) ESC, ILE, I, DNOMB(I), ESC, ILE, VT, ANG, I
          IF (ILE.GT. 9.AND. DISPOS.EQ. 'PA'.AND. KO.EQ. 0)
          *      WRITE(IWR, 632) ESC, ILE, I, DNOMB(I), ESC, ILE, VT, ANG, I

```

Se imprime información nodal por impresora.

```

      IF (DISPOS.EQ.'IM'.AND.KO.EQ.0)
*         WRITE (IWR,633) I, DNOMB(I), VT, ANG, I
      IF (DISPOS.EQ.'IM'.AND.KO.EQ.1)
*         WRITE (IWR,639) I, DNOMB(I), VT, ANG, I
      ILE = ILE + 1
      IF (C.EQ.0.0) GO TO 528
      ES = ' '
      IF (TIPO(I).EQ.3.AND.B.NE.0.0) GO TO 522
      ES = 'R'
522     DO 523 L = 1, NNGEN
          L1 = NPV(L)
          IF (L1.NE.1) GO TO 523
          IF (QG(I).EQ.QGMIN(L)) ES = 'I'
          IF (QG(I).EQ.QGMAX(L)) ES = 'S'
          GO TO 524
523     CONTINUE
524     IF (KO.EQ.0) GO TO 528
      IF (DISPOS.EQ.'IM'.AND.NAU(L).LT.0) WRITE(IWR,644)A, B, ES, C
      IF (DISPOS.EQ.'IM'.AND.NAU(L).GT.0) WRITE(IWR,646)A, B, ES, C
      IF (ILE.LE.9.AND.DISPOS.EQ.'PA'.AND.NAU(L).LT.0)
*         WRITE (IWR,640) ESC, ILE, A, B, ES, C
      IF (ILE.GT.9.AND.DISPOS.EQ.'PA'.AND.NAU(L).LT.0)
*         WRITE (IWR,641) ESC, ILE, A, B, ES, C
      IF (ILE.LE.9.AND.DISPOS.EQ.'PA'.AND.NAU(L).GT.0)
*         WRITE (IWR,642) ESC, ILE, A, B, ES, C
      IF (ILE.GT.9.AND.DISPOS.EQ.'PA'.AND.NAU(L).GT.0)
*         WRITE (IWR,643) ESC, ILE, A, B, ES, C
      ILE = ILE + 1
528     IF (F.EQ.0.0.OR.KO.EQ.0) GO TO 529
      IF (ILE.LE.9.AND.DISPOS.EQ.'PA') WRITE(IWR,650) ESC, ILE, D, E, F
      IF (ILE.GT.9.AND.DISPOS.EQ.'PA') WRITE(IWR,652) ESC, ILE, D, E, F
      IF (DISPOS.EQ.'IM') WRITE (IWR,654) D, E, F
      ILE = ILE + 1
529     IF (G.EQ.0.0.OR.KO.EQ.0) GO TO 531
      IF(QM.LT.0.0.AND.ILE.LE.9.AND.DISPOS.EQ.'PA')
*         WRITE(IWR,660) ESC, ILE, PM, QM, G
      IF(QM.LT.0.0.AND.ILE.GT.9.AND.DISPOS.EQ.'PA')
*         WRITE(IWR,662) ESC, ILE, PM, QM, G
      IF(QM.GT.0.0.AND.ILE.LE.9.AND.DISPOS.EQ.'PA')
*         WRITE(IWR,664) ESC, ILE, PM, QM, G
      IF(QM.GT.0.0.AND.ILE.GT.9.AND.DISPOS.EQ.'PA')
*         WRITE(IWR,666) ESC, ILE, PM, QM, G
      IF(QM.LT.0.0.AND.DISPOS.EQ.'IM') WRITE(IWR,668) PM, QM, G
      IF(QM.GT.0.0.AND.DISPOS.EQ.'IM') WRITE(IWR,670) PM, QM, G
      ILE = ILE + 1

```

Se inicia el proceso de cálculo e impresión de flujos de potencia.
--

```

531     KC = INICPR(I)
535     IF (KC.EQ.0) GO TO 584
          I1 = IPOSPR(KC)

```

```

IF (I1.LT.0) GO TO 580
IF (NE(I1).LT.0) GO TO 580
IC = NR(I1)
YPQ1 = YPQ2E(I1)
YPQ2 = YPQ2R(I1)
GPQ1 = GPQ2E(I1)
GPQ2 = GPQ2R(I1)
IF(I.EQ.NE(I1)) GO TO 554
IC = NE(I1)
YPQ1 = YPQ2R(I1)
YPQ2 = YPQ2E(I1)
GPQ1 = GPQ2R(I1)
GPQ2 = GPQ2E(I1)
554 V3 = VRE(IC)
V4 = VIM(IC)
VJ = V3 * V3 + V4 * V4
A = V1 * V3 + V2 * V4
B = V2 * V3 - V1 * V4
GPQ = R1(I1)
BPQ = X1(I1)
BP2 = - YPQ2 - BPQ
PIJ = (VI * (GPQ + GPQ1) - GPQ * A - B * BPQ) * SBASE
QIJ = (-VI * (YPQ1 + BPQ) + A * BPQ - B * GPQ) * SBASE
PJI = (VJ * (GPQ + GPQ1) - GPQ * A + B * BPQ) * SBASE
QJI = (-VJ * (YPQ2 + BPQ) + A * BPQ + B * GPQ) * SBASE
MP = ( PIJ + PJI ) * 1000.0
MQ = ( QIJ + QJI ) * 1000.0
A = SQRT (MP * MP + MQ * MQ)

```

Se revisa si el elemento de transmisión es transformador.

```

T = 0.0
IF(GPQ.NE.0.0.AND.YPQ2E(I1).EQ.YPQ2R(I1))GO TO 560
IF(YPQ2E(I1).EQ.YPQ2R(I1).AND.YPQ2E(I1).NE.0.0)GO TO 560
DO 555 K7 = 1, NTAPS
T1 = TAP(K7)
N3 = NETR(K7)
N4 = NRTR(K7)
IF (T1.LT.0.0) GO TO 555
IF (N3.NE.I.AND.N4.NE.I) GO TO 555
IF (N3.NE.IC.AND.N4.NE.IC) GO TO 555
T = T1
IF(N3.EQ.IC.AND.N4.EQ.I) T = - T1
TAP(K7) = - T1
GO TO 560
555 CONTINUE

```

Se imprimen flujos.

```

560 PM = SQRT( PIJ * PIJ + QIJ * QIJ )
IF (DISPOS.EQ.'PA') GO TO 570
IF (KO.EQ.0) GO TO 565
IF(T.EQ.0.0)WRITE(IWR,693)IC, DNOMB(IC), PIJ, QIJ, PM
IF(T.NE.0.0)WRITE(IWR,694)IC, DNOMB(IC), PIJ, QIJ, PM, T

```

```

GO TO 575
565 IF(T.EQ.O.O)WRITE(IWR,690)IC, DNOMB(IC), MP, MQ, A
IF(T.NE.O.O)WRITE(IWR,692)IC, DNOMB(IC), MP, MQ, A, T
GO TO 575
570 IF (KO.EQ.O) GO TO 573
IF(T.EQ.O.O.AND.ILE.LE.9)
* WRITE(IWR,680) ESC, ILE, IC, DNOMB(IC), PIJ, QIJ, PM
IF(T.EQ.O.O.AND.ILE.GT.9)
* WRITE(IWR,682) ESC, ILE, IC, DNOMB(IC), PIJ, QIJ, PM
IF(T.NE.O.O.AND.ILE.LE.9)
* WRITE(IWR,684) ESC, ILE, IC, DNOMB(IC), PIJ, QIJ, PM, T
IF(T.NE.O.O.AND.ILE.GT.9)
* WRITE(IWR,686) ESC, ILE, IC, DNOMB(IC), PIJ, QIJ, PM, T
GO TO 575
573 IF(T.EQ.O.O.AND.ILE.LE.9)
* WRITE(IWR,680) ESC, ILE, IC, DNOMB(IC), MP, MQ, A
IF(T.EQ.O.O.AND.ILE.GT.9)
* WRITE(IWR,682) ESC, ILE, IC, DNOMB(IC), MP, MQ, A
IF(T.NE.O.O.AND.ILE.LE.9)
* WRITE(IWR,684) ESC, ILE, IC, DNOMB(IC), MP, MQ, A, T
IF(T.NE.O.O.AND.ILE.GT.9)
* WRITE(IWR,686) ESC, ILE, IC, DNOMB(IC), MP, MQ, A, T

```

Se calculan pérdidas.

```

575 PL = PL + PIJ
QL = QL + QIJ
ILE = ILE + 1
580 KC = IFINPR(KC)
GO TO 535
584 DO 585 K7 = 1, NTAPS
585 TAP(K7) = ABS(TAP(K7))
590 CONTINUE

```

Se inhibe el nodo en cuestión mediante un cambio al signo de su tipo.

```

TIPO(I) = - TIPO(I)
600 CONTINUE

```

Se regresan los valores del vector TIPO(.) a su forma original.

```

DO I = 1, NNODOS
TIPO(I) = ABS(TIPO(I))
END DO

```

Formatos de impresión.

```

631 FORMAT (A, '[', I1, ', ;4H', 'DE', 3X, I3, 1X, A8, 5X, 'KW', 5X, 'KVar', /,
* A, '[', I1, ', ;41H', 'KVA', 5X, 'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)
632 FORMAT (A, '[', I2, ', ;4H', 'DE', 3X, I3, 1X, A8, 5X, 'KW', 5X, 'KVar', /,
* A, '[', I2, ', ;41H', 'KVA', 5X, 'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)
633 FORMAT (/3X, 'DE', 3X, I3, 1X, A8, 5X, 'KW', 5X, 'KVar', 5X, 'KVA', 5X,
* 'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)
635 FORMAT (A, '[', I1, ', ;4H', 'DE', 3X, I3, 1X, A8, 5X, 'MW', 5X, 'MVar', /,
* A, '[', I1, ', ;41H', 'MVA', 5X, 'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)

```

```

637 FORMAT (A, '[', I2, '; 4H', 'DE', 3X, I3, 1X, A8, 5X, 'MW', 5X, 'MVA', '/',
*      A, '[', I2, '; 41H', 'MVA', 5X, 'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)
639 FORMAT (/3X, 'DE', 3X, I3, 1X, A8, 5X, 'MW', 5X, 'MVA', 5X,
*      'TAP', 3X, F6.4, 3X, F8.3, 2X, I3)
640 FORMAT (A, '[', I1, '; 4H', 'COMP. ESTATICO', 5X, F6.1, 1X, F7.1, A1, 1X,
*      F6.1)
641 FORMAT (A, '[', I2, '; 4H', 'COMP. ESTATICO', 5X, F6.1, 1X, F7.1, A1, 1X,
*      F6.1)
642 FORMAT (A, '[', I1, '; 4H', 'GENERADOR', 10X, F6.1, 1X, F7.1, A1, 1X, F6.1)
643 FORMAT (A, '[', I2, '; 4H', 'GENERADOR', 10X, F6.1, 1X, F7.1, A1, 1X, F6.1)
644 FORMAT (3X, 'COMP. ESTATICO', 5X, F6.1, 1X, F7.1, A1, 1X, F6.1)
646 FORMAT (3X, 'GENERADOR', 10X, F6.1, 1X, F7.1, A1, 1X, F6.1)
650 FORMAT (A, '[', I1, '; 4H', 'A CARGA', 11X, F6.1, 1X, F7.1, 2X, F6.1)
652 FORMAT (A, '[', I2, '; 4H', 'A CARGA', 11X, F6.1, 1X, F7.1, 2X, F6.1)
654 FORMAT (4X, 'A CARGA', 10X, F6.1, 1X, F7.1, 2X, F6.1)
660 FORMAT (A, '[', I1, '; 4H', 'A CAPACITOR', 6X, F6.1, 1X, F7.1, 2X, F6.1)
662 FORMAT (A, '[', I2, '; 4H', 'A CAPACITOR', 6X, F6.1, 1X, F7.1, 2X, F6.1)
664 FORMAT (A, '[', I1, '; 4H', 'A REACTOR', 8X, F6.1, 1X, F7.1, 2X, F6.1)
666 FORMAT (A, '[', I2, '; 4H', 'A REACTOR', 8X, F6.1, 1X, F7.1, 2X, F6.1)
668 FORMAT (4X, 'A CAPACITOR', 6X, F6.1, 1X, F7.1, 2X, F6.1)
670 FORMAT (4X, 'A REACTOR', 8X, F6.1, 1X, F7.1, 2X, F6.1)
680 FORMAT (A, '[', I1, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X, F6.1)
682 FORMAT (A, '[', I2, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X, F6.1)
684 FORMAT (A, '[', I1, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X,
*      F6.1, 2X, F7.4, ' TR')
686 FORMAT (A, '[', I2, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X,
*      F6.1, 2X, F7.4, ' TR')
690 FORMAT (4X, 'A', 3X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X, F6.1)
692 FORMAT (4X, 'A', 3X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X, F6.1, 2X, F7.4, ' TR')
693 FORMAT (4X, 'A', 3X, I3, 1X, A8, 1X, F7.5, 1X, F7.5, 1X, F7.5)
694 FORMAT (4X, 'A', 3X, I3, 1X, A8, 1X, F7.5, 1X, F7.5, 1X, F7.5, 2X, F7.4, ' TR')
695 FORMAT (A, '[', I1, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 1X, F7.1)
696 FORMAT (A, '[', I2, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 1X, F7.1)
697 FORMAT (A, '[', I1, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X,
*      F7.1, 2X, F7.4, ' TR')
698 FORMAT (A, '[', I2, '; 4H', 'A', 4X, I3, 1X, A8, 1X, F7.1, 1X, F7.1, 2X,
*      F7.1, 2X, F7.4, ' TR')

```

Se regresa el control para preguntar por otro nodo u otra isla.

700 IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 66

Se expresan los totales de carga, generación y pérdidas en MW y MVars, para cada isla.

```

TOTPG      = TOTPG * SBASE
TOTQG      = TOTQG * SBASE
TOTPC      = TOTPC * SBASE
TOTQC      = TOTQC * SBASE
TOTQG      = TOTQC + QL
TPG(KISLA) = TOTPG
TQG(KISLA) = TOTQG
TPC(KISLA) = TOTPC
TQC(KISLA) = TOTQC

```

```

PLS(KISLA) = PL
QLS(KISLA) = QL
IF (DISPOS.EQ.'PA') GO TO 714

```

Se imprimen los totales de carga, generación y pérdidas.

```

WRITE (IWR,712) TOTPG, TOTQG, TOTPC, TOTQC, PL, QL
712  FORMAT(/25X,'TOTAL DE GENERACION :',
*      /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*      //25X,'TOTAL DE CARGA :',
*      /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*      //25X,'PERDIDAS EN EL SISTEMA :',
*      /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS')
GO TO 718
714  CALL ENCA
CALL CUADRO
WRITE (IWR,382) ESC, ESC, KISLA, ESC
WRITE (IWR,716)ESC,ESC,TOTPG,TOTQG,ESC,ESC,TOTPC,TOTQC,
*      ESC,ESC,PL,QL
716  FORMAT(A,'[8;25H','TOTAL DE GENERACION :',
*      /,A,'[10;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*      /,A,'[13;25H','TOTAL DE CARGA :',
*      /,A,'[15;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*      /,A,'[18;25H','PERDIDAS EN EL SISTEMA :',
*      /,A,'[20;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS')

```

Si la impresión no es por impresora se pide "entrada" para continuar...

```

718  IF (IWR.NE.8) CALL ENCA
ILE = 4
720  CONTINUE

```

Se cierra el archivo de reporte por impresora.

```

IF (IWR.EQ.8.AND.IFLAG.EQ.1) CLOSE (UNIT=8)

```

Se regresa el control al menú de reportes para seleccionar otra opción.

```

IF (IFLAG.EQ.1) GO TO 50
IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 66
CALL CUADRO
WRITE (LTERM,65) ESC, ESC
GO TO 66
737  WRITE(6,739)
739  FORMAT(//2X,'ERROR EN DATOS DE COMPENSADORES !!!')

```

Fin del subprograma FINALE.

```

3000 RETURN
END

```

b) Subprograma DATREP

Las actividades de este subprograma son:

Creación de reportes de reportes de datos de la red de transmisión, ya sea completos, por isla eléctrica o por nodo. Tales reportes pueden enviarse a través de pantalla o un archivo tipo ASCII que pueda ser consultado posteriormente.

Este subprograma es accedido mediante la instrucción:

```
CALL  DATREP  (NNODOS, NLT, R, X, R1, X1, GPQ2E, GPQ2R, YPQ2E, YPQ2R,
*          NSHNTS, NSHUNT, SHUNT, DNOMB, NISLAS, INISLA, ISLA,
*          NAU, NPV, NNGEN, YM, ABORTA, LETR1, LETR2, TIPO)
```

Una descripción de las variables involucradas en el argumento del subprograma, se muestra a continuación..

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NLT(NNODOS)	E	ENTERA	Indica si el nodo asociado a la posición contiene un compensador en derivación: si $NLT(i) = 0$, el nodo i no contiene compensación en derivación; si $NLT(i) = 1$, el nodo i sí la contiene.
R(NLINES)	E	REAL	Resistencia serie del circuito π de cada elemento de transmisión.
X(NLINES)	E	REAL	Reactancia inductiva serie del circuito π de cada elemento de transmisión.
R1(NLINES)	E	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	E	REAL	Susceptancia inductiva serie del circuito π de cada elemento de transmisión.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
YPQ2E(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de envío en el circuito π de cada elemento de transmisión.
YPQ2R(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de recepción en el circuito π del elemento de transmisión.
GPQ2E(NLINES)	E	REAL	Valor de la conductancia de la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de envío.
GPQ2R(NLINES)	E	REAL	Valor de la conductancia de la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de recepción.
NSHNTS	E	ENTERA	Total de compensadores en derivación en el sistema eléctrico de potencia.
NSHUNT(NSHNTS)	E	ENTERA	Indica el nodo en donde se conecta un compensador en derivación.
SHUNT(NSHNTS)	E	REAL	Valor de la susceptancia del compensador en derivación conectado al nodo indicado en NSHUNT(.).
DNOMB(NNODOS)	E	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema eléctrico.
INISLA(NISLAS)	E	ENTERA	Indica el número de nodo en donde se inicia la información topológica de la isla indicada por la posición de INISLA(.).
ISLA(NISLAS)	E	ENTERA	Contiene los nodos del sistema, en forma de listas encadenadas para cada isla.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNGEN	E	ENTERA	Total de nodos generadores en el sistema eléctrico.
NAU(NNGEN)	E	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
NPV(NNGEN)	E	ENTERA	Contiene en cada posición el número correspondiente a un nodo de generación.
YM(NNGEN)	E	REAL	Susceptancia equivalente de compensadores estáticos de potencia reactiva con límites violados.
ABORTA	E/S	ENTERA	Indica si al salir de este subprograma se ejecuta la siguiente instrucción del programa principal (ABORTA = 0) ó se termina la sesión de simulación (ABORTA = 1).
LETR1	E	CHARACTER	Primer renglón descriptivo del archivo de datos del sistema eléctrico.
LETR2	E	CHARACTER	Segundo renglón descriptivo del archivo de datos del sistema eléctrico.
TIPO(NNODOS)	E	ENTERA	Tipo de cada nodo del sistema eléctrico.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica donde se inicia la información referente a la posición de circuitos π , la cual se encuentra en IPOSPR(), correspondiente al nodo i.
NE(NLINES)	E	ENTERA	NE(i) indica el nodo de envío correspondiente al circuito π i-ésimo.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NR(NLINES)	E	ENTERA	NR(i) indica el nodo de recepción correspondiente al circuito π i-ésimo.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(1).
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k) indica donde se encuentra un circuito π relacionado con el nodo i, donde k está definida por IFINPR(.).
NTAPS	E	ENTERA	Total de transformadores con tap variable.
NETR(NTAPS)	E	ENTERA	Identificador del nodo al lado del tap de transformadores con tap variable.
NRTR(NTAPS)	E	ENTERA	Identificador del nodo al lado contrario del tap de transformadores con tap variable.
TAP(NTAPS)	E	REAL	Valor del tap en cada uno de los transformadores con tap variable.
VALNUM(15)	E/S	REAL	Almacena números reales que corresponden a variables involucradas en un menú.
DEPEND(15)	E/S	ENTERA	Indica dependencia entre opciones de un menú.
LISTOP(15)	E/S	CHARACTER	Contiene las cadenas de caracteres describiendo cada opción del menú.
ENCABE	E/S	CHARACTER	Cadena de caracteres alfanuméricos para identificar cada menú.
VALCHA(15)	E/S	CHARACTER	Contiene variables de tipo alfanumérico asociadas con alguna opción del menú.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ALTERN	E/S	CHARACTER	Cadena de caracteres alfanuméricos para formular la pregunta de selección de opción.
TIPVAL(15)	E/S	CHARACTER	Cadena de tres caracteres alfanuméricos indicando el tipo de variable asociada con cada opción.

Las demás variables utilizadas en este subprograma se describen conforme se explica cada parte de código, el cual se presenta enseguida.

Inicio del subprograma DATREP.

```

C
C   REPORTES DE PARAMETROS DEL SISTEMA ELECTRICO.
C
   SUBROUTINE DATREP (NNODOS, NLT, R, X, R1, X1, GPQ2E, GPQ2R, YPQ2E, YPQ2R,
*                       NSHNTS, NSHUNT, SHUNT, DNOMB, NISLAS, INISLA, ISLA,
*                       NAU, NPV, NNGEN, YM, ABORTA, LETR1, LETR2, TIPO)

```

Declaración de bloques comunes y de variables.
--

```

COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
C
   INTEGER NETR(300), NRTR(300), IPOSPR(2000), ABORTA, OPCION, ISLA(550),
*           NSHUNT(150), INICPR(550), IFINPR(2000), INISLA(10), NR(1000),
*           NLT(550), NAU(150), NE(1000), NPV(150), DEPEND(15), TIPO(550)

   REAL R1(1000), X1(1000), GPQ2E(1000), GPQ2R(1000), YPQ2E(1000),
*       R(1000), X(1000), YPQ2R(1000), TAP(300), SHUNT(150), YM(150)
C
   CHARACTER LISTOP(15)*50, ENCABE*50, ALTERN*50, NOMB*8, DNOMB(550)*8
   CHARACTER TIPVAL(15)*3, ARCH*40, NAME*40, NOMB1*8, VALCHA(15)*12
   CHARACTER ESC*1, DISPOS*2, LETR1*70, LETR2*70, WHITEFORE*4, YELFORE*4
   PARAMETER (YELFORE='[33m', WHITEFORE = '[37m')
   REAL*8 VALNUM(15)

```

Se inicia la variable ESC que sirve para crear secuencias de 'escape' en pantalla. LTERM se define como canal de salida por pantalla. Esta se limpia ejecutando el subprograma CLEAR. Al llamar al subprograma INIMEN se inicializa el menú principal de reportes de parámetros.
--

```

ESC = CHAR(27)
LTERM = 6

```

```

60 IWR = LTERM
   CALL CLEAR
   CALL INIMEN

```

Se especifican las opciones del reporte principal de parámetros.

```

ENCABE = 'OPCIONES PARA REPORTE DE PARAMETROS : '
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' REGRESAR A MENU DE REPORTE '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' REPORTE COMPLETO '
LISTOP(4) = ' REPORTE POR NODO '
IF (NISLAS.EQ.1) GO TO 61
LISTOP(5) = ' REPORTE POR ISLA '
61 OPCION = 1

```

Se llama al subprograma MENU donde se define la opción a ejecutar.

```

CALL MENU (OPCION)
IF (OPCION.EQ.3) GO TO 170
IF (OPCION.EQ.4.OR.OPCION.EQ.5) GO TO 63

```

Al cumplirse la siguiente pregunta se sale de este menú de reportes.

```

IF (OPCION.EQ.1) GO TO 3000

```

Si OPCION = 2, se regresa al menú principal de reportes.

```

IF (OPCION.EQ.2) GO TO 60

```

Se ha seleccionado la opción de terminación del programa (OPCION = 2).

```

ABORTA = 1
RETURN

```

Reporte de parámetros por nodo o por isla.

```

63 CONTINUE

```

KISLA es el número de isla por reportar, iniciándose en 0. IFLAG = 0, indica reporte por nodo. IFLAG = 1 indica reporte completo.

```

KISLA = 0
IFLAG = 0
NOMB1 = 'NODO'
IF (OPCION.EQ.5) NOMB1 = 'ISLA'

```

Se borra pantalla, se imprime encabezado de la opción seleccionada y se pregunta ya sea por el nombre del nodo o por el número de isla.

```

CALL CUADRO
WRITE (LTERM,64) ESC, NOMB1, ESC
64 FORMAT (A,'[3;25H','REPORTE DE PARAMETROS POR ',A4,/,
*      A,'[4;25H','----- -- ----- ---- ----')
NOMB1 = ' '
GO TO 175
66 WRITE (LTERM,69) ESC

```

```

IF (OPCION.EQ.4) WRITE (LTERM,70) ESC, ESC
IF (OPCION.EQ.5) WRITE (LTERM,72) ESC, NISLAS, ESC
69 FORMAT (A,'[24;4H','
*
')
70 FORMAT (A,'[24;6H','¿ QUE NODO [maximo 8 caracteres ,
*
return para salir] ?',' _____',A,'[24;61H',,$)
72 FORMAT (A,'[24;6H','¿ CUAL ISLA [ num. entero entre 1 y ',I2,
*
' return para salir ] ? ',/,A,'[24;73H',,$)

```

Aquí se lee nombre del nodo o el número de isla.

```

IF (OPCION.EQ.4) READ (LTERM,80) NOMB1
IF (OPCION.EQ.5) READ (LTERM,81) KISLA
WRITE (LTERM,78) ESC
78 FORMAT (A,'[16;10H','
')
80 FORMAT(A8)
81 FORMAT(I2)

```

Se pasa la primera prueba para lograr un reporte por nodo.

```

IF (OPCION.EQ.4.AND.NOMB1.NE.'
')GO TO 85

```

Se pasa la primera prueba para lograr un reporte por isla o sistema.

```

IF (OPCION.EQ.5.AND.KISLA.NE.0) GO TO 83

```

Se cierra canal de escritura al no haber más reportes por impresora.

```

IF (NOMB.EQ.'IM') CLOSE (UNIT = 8)
GO TO 60

```

Se verifica la existencia de la isla proporcionada por el usuario.

```

83 IF (KISLA.LE.NISLAS) GO TO 340
WRITE (*,84) ESC, KISLA
84 FORMAT (A,'[19;10H',' ** LA ISLA ',I2,' NO EXISTE ')
GO TO 66

```

Se verifica la existencia del nodo proporcionado por el usuario.

```

85 DO 90 J = 1, NNODOS
IF(NOMB1.EQ.DNOMB(J))GO TO 150
90 CONTINUE
WRITE (LTERM,100) ESC, NOMB1
100 FORMAT(A,'[19;10H',' ** EL NODO ',A8,' NO EXISTE ')
102 FORMAT (A,'[19;10H',' ** EL NODO ',A8,' FUE REDUCIDO ')
IFLAG = 1
GO TO 66

```

Se revisa si el nodo no fué reducido por ser un CEV con límite violado.

```

150 DO 154 I1 = 1, NNODOS
IF (ISLA(I1).EQ.J) GO TO 160
IF (ABS(ISLA(I1)).EQ.J) GO TO 155
154 CONTINUE
155 WRITE(6,102) I1, DNOMB(I1)

```

```

GO TO 66
160 CONTINUE

```

Por pantalla se envía mensaje de grabación de archivo para impresora.

```

IF (NOMB.EQ.'IM') WRITE (LTERM,161) ESC, NOMB1
161 FORMAT(A,'{19;10H',' SE GRABO INFORMACION DEL NODO ',A8)
IFLAG = 0
ILE = 9
ISIZE = 25

```

Se limpia la parte de pantalla usada para preguntar por nodo o isla y dispositivo de impresión de reporte.

```

DO 167 KJ = 1, LIMPIA
DO 166 J = 2, 18
IF (J.LT.10) WRITE (LTERM,163) ESC, J, ESC, J
IF (J.GE.10) WRITE (LTERM,164) ESC, J, ESC, J
163 FORMAT (A,'{',I1,';3H',' ',/,A,'{',
*I1,';27H',' ')
164 FORMAT (A,'{',I2,';3H',' ',/,A,'{',
*I2,';27H',' ')
166 CONTINUE
167 CONTINUE
DO 168 J = 19, 22
168 WRITE (LTERM,164) ESC, J, ESC, J
WRITE (LTERM,169) ESC
169 FORMAT (A,'{7;4H',' ',/,
*
ILE = 1
GO TO 340
170 CONTINUE

```

Aquí, se efectúan reportes completos ya sea por pantalla o impresora.

```

IFLAG = 1
CALL CUADRO
WRITE (LTERM,172) ESC, ESC
172 FORMAT (A,'{4;25H',' REPORTE DE PARAMETROS COMPLETO ',/,
* A,'{5;25H','----- -- ----- ')

```

Aquí se define el dispositivo de salida para reporte de datos.

```

175 NOMB = 'PA'
WRITE (LTERM, 180) ESC, ESC , YELFORE
WRITE (LTERM, 190) ESC, ESC, WHITEFORE, NOMB, ESC
180 FORMAT (A,'{10;14H', A, A4,
* '¿ REPORTE EN IMPRESORA O PANTALLA [IM, PA] ? ')
190 FORMAT (A,'{10;69H',A,A4,A2,A,'{10;69H',A)
READ(LTERM,200,ERR=210)NOMB
200 FORMAT(A2)
WRITE (LTERM, 202) ESC, ESC , YELFORE
WRITE (LTERM, 205) ESC, ESC, WHITEFORE, ESC
202 FORMAT (A,'{10;14H', A, A4,
*

```

```

205 FORMAT (A, '[10;69H', A, A4, ' ', A, '[10;69H', '$)
      IF(NOMB.EQ. 'IM'.OR.NOMB.EQ. 'im')GO TO 240
      IF(NOMB.EQ. 'PA'.OR.NOMB.EQ. 'pa'.OR.NOMB.EQ. ' ') GO TO 320
210 CONTINUE
      WRITE(LTERM, 220)
220 FORMAT(//10X, ' * RESPUESTA INCORRECTA, FORMATO A2')
      GO TO 170
240 CONTINUE

```

Se ha definido como dispositivo de salida la impresora. Ahora, se se define el nombre del archivo a donde será guardado el reporte.

```

DISPOS = 'IM'
LIMPIA = 2
ARCH = 'DATOS.SAL'
WRITE (LTERM, 250) ESC, ESC, YELFORE
WRITE (LTERM, 260) ESC, ESC, WHITEFORE, ARCH, ESC
250 FORMAT (A, '[15;3H', A, A4, '¿ NOMBRE DEL ARCHIVO PARA REPORTE ? ')
260 FORMAT (A, '[15;40H', A, A4, A12, A, '[15;40H', '$)
      READ(LTERM, 270)NAME
270 FORMAT(A40)
      IF (NAME.EQ. ' ') GO TO 280
      ARCH = NAME

```

El canal de escritura de archivo es el 8, y se abre el archivo con el nombre definido anteriormente.

```

280 IWR = 8
      OPEN (UNIT = 8, FILE = ARCH )
      ISIZE = 50

```

Se imprime el encabezado del archivo de datos del sistema eléctrico.

```

WRITE(IWR, 300)LETR1, LETR2
300 FORMAT(/1X, A70, /, 1X, A70)
      NOMB = 'IM'
      IF (IFLAG.EQ.0) GO TO 66
      GO TO 340
320 CONTINUE

```

El dispositivo de salida es pantalla y se define el tamaño de la misma

```

DISPOS = 'PA'
LIMPIA = 1
ISIZE = 21
IWR = LTERM
IF (IFLAG.EQ.0) GO TO 66

```

Se define espacio en regiones para imprimir el número de isla que se está reportando. Si existe sólo una isla no se imprime tal letrero.

```

340 IF (NISLAS.EQ.1) ILE = 1
      IF (NISLAS.GT.1) ILE = 4
371 IANCHO = 23
      IRENGL = 2
      IF (NISLAS.GT.1) IRENGL = 5

```



```

IF (NISLAS.GT.1) IANCHO = 22
INIC1 = 1
IND = 0
IFIN1 = NISLAS
IF (IFLAG.NE.0) GO TO 372

```

Se define el número de nodos a imprimir. En este caso, el reporte es por nodo, por lo que sólo se imprime referente al mismo.

```

INICIO = 1
IFIN1 = 1
IFIN = 1
IF (OPCION.NE.5) GO TO 372

```

Se definen límites del ciclo iterativo de la isla a imprimir.

```

INIC1 = KISLA
IFIN1 = KISLA

```

Ciclo de impresión.

```

372 DO 620 KISLA = INIC1, IFIN1
    IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 384
    IF (DISPOS.EQ.'IM'.AND.NISLAS.GT.1) WRITE(IWR,380) KISLA
    IF (DISPOS.EQ.'PA') CALL CUADRO
    IF (DISPOS.EQ.'PA'.AND.NISLAS.GT.1)WRITE(*,382)ESC,ESC,KISLA,ESC
380  FORMAT(//20X,'*****',
*      /20X,'PARAMETROS DE LA ISLA No. ',I2,
*      /20X,'*****',//)
382  FORMAT(A,'[2;20H','*****',/,
*      A,'[3;20H','PARAMETROS DE LA ISLA No. ',I2,/,
*      A,'[4;20H','*****')
    INICIO = INISLA(KISLA)
    IF (KISLA.LT.NISLAS) IFIN = INISLA(KISLA + 1) - 1
    IF (KISLA.EQ.NISLAS) IFIN = NNODOS

```

Se imprime el ciclo para cada nodo.

```

384  DO 500 JS = INICIO, IFIN
    I = ISLA(JS)
    IF (I.LT.0) GO TO 500
    QM = 0.0

```

Se verifica información de compensación en derivación.

```

    IF(NLT(I).EQ.0) GO TO 440
    IS = 0
436  IS = IS + 1
    IF(IS.GT.NSHNTS)GO TO 637
    IF(NSHUNT(IS).NE.I) GO TO 436
    QM = SHUNT(IS)
    ILE = ILE + 1

```

KC es la posición que guarda cada nodo en la lista representando una isla. KC3 contabiliza los renglones que se imprimen en pantalla.

```

440      KC = INICPR(I)
          KC3 = 1
441      KC = IFINPR(KC)
          IF (KC.EQ.0) GO TO 442
          KC3 = KC3 + 1
          GO TO 441
442      KC3 = KC3 + ILE

```

Se reserva un renglón para imprimir datos de compensación en derivación

```

          IF (QM.NE.0.0) ILE = ILE - 1
          ILE = ILE + 1
          IF (KC3.LE.IANCHO.OR.DISPOS.EQ.'IM') GO TO 444

```

Se termina una página, por lo que se inicia una nueva.

```

          ILE = IRENGL
          CALL ENCA
          CALL CUADRO
          IF (OPCION.EQ.3.AND.NISLAS.GT.1)WRITE(*,382)ESC,ESC,KISLA,ESC
          IF (OPCION.EQ.5.AND.NISLAS.GT.1)WRITE(*,382)ESC,ESC,KISLA,ESC

```

Se imprime letrero para valores de elementos incidentes al nodo actual

```

444      IF (ILE.LE.9.AND.DISPOS.EQ.'PA')
*          WRITE(IWR,535) ESC, ILE, I, DNOMB(I), ESC, ILE
          IF (ILE.GT.9.AND.DISPOS.EQ.'PA')
*          WRITE(IWR,537) ESC, ILE, I, DNOMB(I), ESC, ILE
          IF (DISPOS.EQ.'IM') WRITE(IWR,539) I, DNOMB(I)
          ILE = ILE + 1
449      IF (QM.EQ.0.0) GO TO 451

```

Impresión de datos de compensación en derivación.

```

*          IF(QM.GT.0.0.AND.ILE.LE.9.AND.DISPOS.EQ.'PA')
*              WRITE(IWR,560) ESC, ILE, QM
*          IF(QM.GT.0.0.AND.ILE.GT.9.AND.DISPOS.EQ.'PA')
*              WRITE(IWR,562) ESC, ILE, QM
*          IF(QM.LT.0.0.AND.ILE.LE.9.AND.DISPOS.EQ.'PA')
*              WRITE(IWR,564) ESC, ILE, QM
*          IF(QM.LT.0.0.AND.ILE.GT.9.AND.DISPOS.EQ.'PA')
*              WRITE(IWR,566) ESC, ILE, QM
          IF(QM.GT.0.0.AND.DISPOS.EQ.'IM') WRITE(IWR,568) QM
          IF(QM.LT.0.0.AND.DISPOS.EQ.'IM') WRITE(IWR,570) QM
          ILE = ILE + 1
451      KC = INICPR(I)

```

Se verifica si terminó la lista encadenada correspondiente al nodo I.

```

453      IF (KC.EQ.0) GO TO 484

```

J es el siguiente nodo que se interconecta al actual. Se verifica que exista en el sistema. Si J < 0, entonces ya no existe.

```

          J = IPOSPR(KC)
          IF (J.LT.0) GO TO 480

```

```

IC = NE(J)
IF (IC.LT.0) GO TO 480

```

El nodo J existe. Ahora, se revisa si el nodo I es el nodo de envío.

```

IF (NE(J).EQ.I) IC = NR(J)
G2E = GPQ2E(J)
G2R = GPQ2R(J)
Y2E = YPQ2E(J)
Y2R = YPQ2R(J)
GPQ = R1(J)
BPQ = X1(J)
RPQ = R(J)
XPQ = X(J)
T = 0.0

```

Se verifica si el elemento de transmisión es línea o transformador.

```

IF (GPQ.NE.0.0.AND.Y2E.EQ.Y2R) GO TO 460
IF (Y2E.EQ.Y2R.AND.Y2E.NE.0.0) GO TO 460
DO 455 K7 = 1, NTAPS
  T1 = TAP(K7)
  N3 = NETR(K7)
  N4 = NRTR(K7)
  IF (T1.LT.0.0) GO TO 455
  IF (N3.NE.I.AND.N4.NE.I) GO TO 455
  IF (N3.NE.IC.AND.N4.NE.IC) GO TO 455
  T = T1

```

Se cambia signo del tap del transformador actual para que no se accese nuevamente en caso de estar en paralelo con otros transformadores.

```

IF(N3.EQ.IC.AND.N4.EQ.I) T = - T1
TAP(K7) = - T1
GO TO 460
455 CONTINUE
460 IND = 0

```

Se imprimen datos del elemento de transmisión, según el caso.

```

IF (ABS(BPQ).GT.1000.0.OR.GPQ.GT.1000.0) IND = 1
IF (DISPOS.EQ.'PA') GO TO 470
IF(T.EQ.0.0.AND.IND.EQ.0)
*   WRITE(IWR,590) IC, DNOMB(IC), Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.EQ.0.0.AND.IND.EQ.1)
*   WRITE(IWR,591) IC, DNOMB(IC), Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.NE.0.0.AND.IND.EQ.0)
*   WRITE(IWR,592) IC, DNOMB(IC), Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
IF(T.NE.0.0.AND.IND.EQ.1)
*   WRITE(IWR,593) IC, DNOMB(IC), Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
GO TO 475
470 IF(T.EQ.0.0.AND.ILE.LE.9.AND.IND.EQ.0)
*   WRITE(IWR,580) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*   Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.EQ.0.0.AND.ILE.LE.9.AND.IND.EQ.1)

```

```

*          WRITE(IWR, 581) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.EQ.O.O.AND.ILE.GT.9.AND.IND.EQ.O)
*          WRITE(IWR, 582) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.EQ.O.O.AND.ILE.GT.9.AND.IND.EQ.1)
*          WRITE(IWR, 583) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ
IF(T.GT.O.O.AND.ILE.LE.9.AND.IND.EQ.O)
*          WRITE(IWR, 584) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
IF(T.GT.O.O.AND.ILE.LE.9.AND.IND.EQ.1)
*          WRITE(IWR, 585) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
IF(T.GT.O.O.AND.ILE.GT.9.AND.IND.EQ.O)
*          WRITE(IWR, 586) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
IF(T.GT.O.O.AND.ILE.GT.9.AND.IND.EQ.1)
*          WRITE(IWR, 587) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2E, Y2R, RPQ, XPQ, GPQ, BPQ, T
IF(T.LT.O.O.AND.ILE.LE.9.AND.IND.EQ.O)
*          WRITE(IWR, 584) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2R, Y2E, RPQ, XPQ, GPQ, BPQ, T
IF(T.LT.O.O.AND.ILE.LE.9.AND.IND.EQ.1)
*          WRITE(IWR, 585) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2R, Y2E, RPQ, XPQ, GPQ, BPQ, T
IF(T.LT.O.O.AND.ILE.GT.9.AND.IND.EQ.O)
*          WRITE(IWR, 586) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2R, Y2E, RPQ, XPQ, GPQ, BPQ, T
IF(T.LT.O.O.AND.ILE.GT.9.AND.IND.EQ.1)
*          WRITE(IWR, 587) ESC, ILE, IC, DNOMB(IC), ESC, ILE,
*          Y2R, Y2E, RPQ, XPQ, GPQ, BPQ, T
475      ILE = ILE + 1
480      KC = IFINPR(KC)
      GO TO 453
484      DO 485 K7 = 1, NTAPS
485          TAP(K7) = ABS(TAP(K7))
500      CONTINUE
535      FORMAT(A, '[', I1, ';3H', 'DE', 1X, I3, 1X, A8, 2X, 'B-env/2 B-rec/2', /,
*          A, '[', I1, ';38H', 'R-serie X-serie G-serie B-serie TAP ')
537      FORMAT(A, '[', I2, ';3H', 'DE', 1X, I3, 1X, A8, 2X, 'B-env/2 B-rec/2', /,
*          A, '[', I2, ';38H', 'R-serie X-serie G-serie B-serie TAP ')
539      FORMAT(/1X, 'DE', 1X, I3, 1X, A8, 2X, 'B-env/2 B-rec/2',
*          ' R-serie X-serie G-serie B-serie TAP ')
560      FORMAT (A, '[', I1, ';3H', 'A CAPACITOR ', 13X, F7.4)
562      FORMAT (A, '[', I2, ';3H', 'A CAPACITOR ', 13X, F7.4)
564      FORMAT (A, '[', I1, ';3H', 'A REACTOR ', 15X, F7.4)
566      FORMAT (A, '[', I2, ';3H', 'A REACTOR ', 15X, F7.4)
568      FORMAT (4X, 'A CAPACITOR', 15X, F7.4)
570      FORMAT (4X, 'A REACTOR', 17X, F7.4)
580      FORMAT (A, '[', I1, ';3H', 'A', 2X, I3, 1X, A8, 1X, /,
*          A, '[', I1, ';20H', 4(F7.4, 2X), F7.3, 2X, F7.2)
581      FORMAT (A, '[', I1, ';3H', 'A', 2X, I3, 1X, A8, 1X, /,

```

```

*      A, '[' , I1, ' ; 20H' , 4(F7.4,2X) , F7.2, 1X, F8.2)
582  FORMAT (A, '[' , I2, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I2, ' ; 20H' , 4(F7.4,2X) , F7.3, 2X, F7.2)
583  FORMAT (A, '[' , I2, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I2, ' ; 20H' , 4(F7.4,2X) , F7.2, 1X, F8.2)
584  FORMAT (A, '[' , I1, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I1, ' ; 20H' , 4(F7.4,2X) , F7.3, 2X, F7.2, 1X, F6.3)
585  FORMAT (A, '[' , I1, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I1, ' ; 20H' , 4(F7.4,2X) , F7.2, 1X, F8.2, 1X, F6.3)
586  FORMAT (A, '[' , I2, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I2, ' ; 20H' , 4(F7.4,2X) , F7.2, 2X, F7.2, 1X, F6.3)
587  FORMAT (A, '[' , I2, ' ; 3H' , 'A' , 2X, I3, 1X, A8, 1X, /,
*      A, '[' , I2, ' ; 20H' , 4(F7.4,2X) , F7.2, 1X, F8.2, 1X, F6.3)
590  FORMAT(2X, 'A' , 1X, I3, 1X, A8, 2X, 4(F7.4,2X) , F7.3, 2X, F7.2)
591  FORMAT(2X, 'A' , 1X, I3, 1X, A8, 2X, 3(F7.4,2X) , F7.4, 1X, F8.3, 1X, F8.3)
592  FORMAT(2X, 'A' , 1X, I3, 1X, A8, 2X, 4(F7.4,2X) , F7.3, 2X, F7.2, 1X, F6.3)
593  FORMAT(2X, 'A' , 1X, I3, 1X, A8, 2X, 3(F7.4,2X) , F7.3, 1X, F8.3, 1X, F8.3,
*      1X, F6.3)

```

Si se trata de reporte por nodo se regresa a la etiqueta 66.

```

IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 66
IF (IWR.NE.8) CALL ENCA
ILE = 4

```

620 CONTINUE

Termina ciclo de impresión de reporte.

```

IF (IWR.EQ.8.AND.IFLAG.EQ.1) CLOSE (UNIT=8)

```

Se regresa a menú de reportes de datos.

```

IF (IFLAG.EQ.1) GO TO 60
IF (IFLAG.EQ.0.AND.OPCION.EQ.4) GO TO 66
CALL CUADRO
NOMB1 = 'ISLA'
WRITE (LTERM,64) ESC, NOMB1, ESC
GO TO 66

```

Se imprime mensaje de error en datos de compensadores.

```

637 WRITE(6,639)
639 FORMAT(//2X,'ERROR EN DATOS DE COMPENSADORES !!!!')

```

Se termina el subprograma DATREP.

```

3000 RETURN
END

```

C.) Subprograma RESREP

Las actividades de este subprograma son:

Creación de reportes resumidos de nodos de carga, generadores, compensación en derivación y balance generación-carga-pérdidas por isla de un sistema eléctrico de potencia. Tales reportes pueden enviarse a través de pantalla o un archivo tipo ASCII que pueda ser consultado posteriormente.

Este subprograma es accesado mediante la instrucción:

```
CALL RESREP (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, R, X, YPQ2E, YPQ2R, VRE,
*          VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT, SHUNT, DNOMB, NISLAS,
*          INISLA, ISLA, NAU, NPV, NNGEN, YM, ABORTA, LETR1, LETR2, QGMIN,
*          QGMAX, KFUERA, NODFUE, ISLACK, GPQ2E, GPQ2R, VOLT, ANGULO,
*          TIPGEN, INIC, ICOL, YBUSR, YBUSI, ISP, LONG, OPCION)
```

Una descripción de las variables del argumento y los bloques comunes involucradas en este subprograma se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
PG(NNODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema.
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.
PC(NNODOS)	E	REAL	Potencia real de carga en cada nodo del sistema.
QC(NNODOS)	E	REAL	Potencia reactiva de carga en cada nodo del sistema.
TIPO(NNODOS)	E	ENTERA	Tipo de cada nodo del sistema eléctrico.
NLT(NNODOS)	E	ENTERA	Indica si el nodo asociado a la posición contiene un compensador en derivación: si $NLT(i) = 0$, el nodo i no contiene compensación en derivación; si $NLT(i) = 1$, el nodo i sí la contiene.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
R1(NLINES)	E	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	E	REAL	Susceptancia inductiva serie del circuito π en elementos de transmisión.
YPQ2E(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de envío en el circuito π de cada elemento de transmisión.
YPQ2R(NLINES)	E	REAL	Mitad del efecto capacitivo asignado al nodo de recepción en el circuito π de cada elemento de transmisión.
VRE(NNODOS)	E	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	E	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
PCAL(NNODOS)	E/S	REAL	Pot. real neta inyectada en cada nodo del sistema.
QCAL(NNODOS)	E/S	REAL	Potencia reactiva neta inyectada en cada nodo del sistema eléctrico.
SBASE	E	REAL	Potencia base del sistema eléctrico en MVA.
NSHNTS	E	ENTERA	Total de compensadores en derivación en el sistema.
NSHUNT(NSHNTS)	E	ENTERA	Indica el nodo en donde se conecta un compensador en derivación.
SHUNT(NSHNTS)	E	REAL	Valor de la susceptancia del compensador en derivación conectado al nodo indicado en NSHUNT(.).
DNOMB(NNODOS)	E	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema eléctrico.
INISLA(NISLAS)	E	ENTERA	Indica el número de nodo en donde se inicia la información topológica de la isla indicada por la posición de INISLA(.).
ISLA(NNODOS)	E	ENTERA	Contiene los nodos del sistema, formando una lista encadenada para cada isla.
NNGEN	E	ENTERA	Total de nodos generadores en el sistema eléctrico.
NAU(NNGEN)	E	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
NPV(NNGEN)	E	ENTERA	Contiene en cada posición el número correspondiente a un nodo de generación.
YM(NNGEN)	E	REAL	Susceptancia equivalente de compensadores estáticos de potencia reactiva con límites violados.
ABORTA	E/S	ENTERA	Indica si al salir de este subprograma se ejecuta la siguiente instrucción del programa principal (ABORTA = 0) ó se termina la sesión de simulación (ABORTA = 1).
LETR1	E	CHARACTER	Primer renglón descriptivo del archivo de datos del sistema eléctrico.
LETR2	E	CHARACTER	Segundo renglón descriptivo del archivo de datos del sistema eléctrico.
QGMIN(NNGEN)	E	REAL	Límite mínimo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
QGMAX(NNGEN)	E	REAL	Limite máximo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).
KFUERA	E	ENTERA	Total de nodos que han sido desconectados del sistema en una sesión del subprograma TOPCAM.
NODFUE(KFUERA)	E	ENTERA	Indica el número de nodo desconectado del sistema en una sesión del subprograma TOPCAM.
ISLACK(NISLAS)	E	ENTERA	Indica el número de nodo asignado como compensador en cada isla eléctrica.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
GPQ2E(NLINES)	E	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de envío.
GPQ2R(NLINES)	E	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de recepción.
VOLT(NNODOS)	E	REAL	Valor de la magnitud del voltaje complejo en cada nodo del sistema eléctrico.
ANGULO(NNODOS)	E	REAL	Valor del ángulo de fase en grados del voltaje complejo en cada nodo del sistema eléctrico de potencia.
INIC(NNODOS)	E	ENTERA	Indica donde se inicia la información de cada nodo de la matriz de admitancias nodal en un esquema de empaquetamiento No. 2.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
TIPGEN(NNGEN)	E	ENTERA	Tipo de nodo de voltaje controlado. Si TIPGEN(k)=0, el nodo de generación k es una máquina síncrona. Si TIPGEN(k)=1, el nodo k es un compensador estático de potencia reactiva.
ICOL(LONG)	E	ENTERA	Contiene información de la columna relacionada al renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal.
LONG	E	ENTERA	LONG = NNODOS + 2*NLINES
YBUSR(LONG)	E	REAL	Parte real de un elemento de la matriz de admitancias nodal.
YBUSI(LONG)	E	REAL	Parte imaginaria de un elemento de la matriz de admitancias nodal.
ISP(LONG)	E	ENTERA	Indica la siguiente posición en la información empaquetada donde se encuentra otro elemento del mismo renglón.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica donde se inicia la información referente a la posición de circuitos π , la cual se encuentra en IPOSPR(), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	NE(i) indica el nodo de envío correspondiente al circuito π i-ésimo.
DEPEND(15)	E/S	ENTERA	Indica dependencia entre opciones de un menú.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NR(NLINES)	E	ENTERA	NR(i) indica el nodo de recepción correspondiente al circuito π i-ésimo.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k) indica donde se encuentra un circuito π relacionado con el nodo i, donde k está definida por IFINPR(.).
VALNUM(15)	E/S	REAL	Almacena números reales que corresponden a variables involucradas en un menú.
LISTOP(15)	E/S	CHARACTER	Tiene cadenas de caracteres alfanuméricos describiendo cada opción del menú.
ENCABE	E/S	CHARACTER	Cadena de caracteres alfanuméricos para identificar cada menú.
ALTERN	E/S	CHARACTER	Cadena de caracteres para formular la pregunta de selección de opción.
TIPVAL(15)	E/S	CHARACTER	Cadena de tres caracteres alfanuméricos indicando el tipo de variable asociada con cada opción.
VALCHA(15)	E/S	CHARACTER	Contiene variables de tipo alfanumérico asociadas con alguna opción del menú.

Las demás variables utilizadas en este subprograma se describen conforme se explica cada parte de código, el cual se presenta enseguida.

Inicio del subprograma RESREP.

C
C
C

REPORTES RESUMIDOS DEL ESTUDIO DE FLUJOS.

```

SUBROUTINE RESREP (NNODOS, PG, QG, PC, QC, TIPO, NLT, R1, X1, R, X, YPQ2E,
* YPQ2R, VRE, VIM, PCAL, QCAL, SBASE, NSHNTS, NSHUNT,
* SHUNT, DNOMB, NISLAS, INISLA, ISLA, NAU, NPV, NNGEN,
* YM, ABORTA, LETR1, LETR2, QGMIN, QGMAX, KFUERA,
* NODFUE, ISLACK, GPQ2E, GPQ2R, VOLT, ANGULO, TIPGEN,
* INIC, ICOL, YBUSR, YBUSI, ISP, LONG, OPCION)

```

Se definen bloques comunes y variables.

```
COMMON /MENU12/ VALNUM,DEPEND,LISTOP,ENCABE,ALTERN,TIPVAL,VALCHA
COMMON /CIRCU1/ NLines, INICPR, IFINPR, NE, NR, IPOSPR
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
```

C

```
INTEGER NE(1000),NR(1000),NSHUNT(150),INICPR(550),NAU(150),
* ISLA(550),NLT(550),IFINPR(2000),IPOSPR(2000),ICOL(2550),
* INISLA(10),NPV(150),TIPO(550),ABORTA,DEPEND(15),OPCION,
* ISLACK(10),TIPGEN(150),INIC(550),ISP(2550),NETR(300),
* NRTR(300),NODFUE(15)
```

C

```
REAL PG(550),QG(550),PC(550),QC(550),VRE(550),VIM(550),R(1000),
* X(1000),YPQ2E(1000),R1(1000),X1(1000),PL(10),YBUSR(2550),
* PCAL(550),QCAL(550),QL(10),TOTPG(10),YM(150),SHUNT(150),
* TOTQG(10),TOTPC(10),TOTQC(10),YPQ2R(1000),YBUSI(2550),
* VOLT(550),ANGULO(550),QGMIN(150),QGMAX(150),ACDER(10),
* TAP(300),GPQ2E(1000),GPQ2R(1000)
```

C

```
CHARACTER LISTOP(15)*50, ENCABE*50, ALTERN*50, DNOMB(550)*8
CHARACTER ESC*1, INVERS*3, DISPOS*2, LETR1*70, TIPVAL(15)*3
CHARACTER WHITEFORE*4, YELFORE*4, LETR2*70
CHARACTER VALCHA(15)*12, ARCH*40, NAME*40, NOMB*8, NOMB1*8
PARAMETER (YELFORE='[33m', WHITEFORE = '[37m')
REAL*8 VALNUM(15)
```

CHAR(27) en Código ASCII crea el efecto de la tecla ESC.

```
ESC = CHAR(27)
```

Se inicializan variables para almacenar totales por isla de:
 potencias real y reactiva de generación: TOTPG(.), TOTQG(.),
 potencias real y reactiva de carga: TOTPC(.), TOTQC(.),
 potencias real y reactiva de pérdidas: PL(.), QL(.),
 aportaciones em MVAr de compensadores en derivación: ACDER(.).

```
DO KISLA = 1, NISLAS
  TOTPG(KISLA) = 0.0
  TOTQG(KISLA) = 0.0
  TOTPC(KISLA) = 0.0
  TOTQC(KISLA) = 0.0
  PL(KISLA) = 0.0
  QL(KISLA) = 0.0
  ACDER(KISLA) = 0.0
```

```
END DO
```

Se define canal de salida por pantalla.

```
LTERM = 6
IWR = LTERM
```

Si OPCION=5, se grabará archivo de flujos para programa de estabilidad.

```
IF (OPCION.EQ.5) GO TO 2700
```

Se inicializan variables para menú en INIMEN y se define el menú para reportes resumidos.

```

29 CALL INIMEN
ENCABE = ' OPCIONES EN REPORTES RESUMIDOS DE FLUJOS : '
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' REGRESAR A SELECTOR DE REPORTES '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' RESUMEN DE GENERADORES '
LISTOP(4) = ' RESUMEN DE CAPACITORES/REACTORES '
LISTOP(5) = ' RESUMEN DE NODOS DEL SISTEMA '
LISTOP(6) = ' RESUMEN CARGA/GEN. DEL SISTEMA '

```

Si existe sólo una isla la definición de menú es la anterior. La siguiente es válida para más de una isla en el sistema.

```

IF (NISLAS.GT.1) GO TO 35
LISTOP(7) = ' IMPRESION DE RESUMEN COMPLETO '
IND = 7
GO TO 40
35 LISTOP(6) = ' RESUMEN DE NODOS POR ISLA '
LISTOP(7) = ' RESUMEN CARGA/GENERACION POR ISLA '
LISTOP(8) = ' IMPRESION DE RESUMEN COMPLETO '
IND = 8
40 OPCION = 1
CALL MENU (OPCION)

```

Si OPCION = 1, se sale de este subprograma regresando al menú de reportes principal. Si OPCION = 2, se sale del simulador de flujos.

```

IF (OPCION.EQ.1) GO TO 3000
IF (OPCION.EQ.2) GO TO 2999
IF (OPCION.LE.6) GO TO 42
IF (OPCION.EQ.7.AND.NISLAS.GT.1) GO TO 42

```

Se define nombre de 'default' para archivo completo de resúmenes.

```

ARCH = 'RESUMEN.SAL'
GO TO 53

```

Se define la pantalla como primera opción de dispositivo de salida. Además, se pregunta al usuario por cuál dispositivo saldrá el reporte.

```

42 NOMB = 'PA'
CALL CUADRO
WRITE (LTERM, 43) ESC, ESC, YELFORE
WRITE (LTERM, 44) ESC, ESC, WHITEFORE, NOMB, ESC
43 FORMAT (A, '[10;14H', A, A4,
* '¿ REPORTE EN IMPRESORA O PANTALLA [IM, PA] ? ')
44 FORMAT (A, '[10;69H', A, A4, A2, A, '[10;69H', '$)

```

Se lee el nombre del dispositivo.

```

READ (LTERM, 45) NOMB
45 FORMAT(A2)

```

```

WRITE (LTERM, 46) ESC, ESC , YELFORE
WRITE (LTERM, 47) ESC, ESC, WHITEFORE, ESC
46 FORMAT (A, '[10;14H', A, A4,
*
47 FORMAT (A, '[10;69H', A, A4, ' ', A, '[10;69H', $)

```

Se verifica la validez de la respuesta. Si se ha definido correctamente a la impresora como dispositivo (NOMB = 'IM'), se va definir archivo. En caso de que NOMB = 'PA', se va directamente a generar el resumen.

```

IF(NOMB.EQ.'IM'.OR.NOMB.EQ.'im') GO TO 50
IF(NOMB.EQ.'PA'.OR.NOMB.EQ.'pa'.OR.NOMB.EQ.' ') GO TO 371

```

La respuesta no es Impresora o PAntalla, por lo que se regresa a efectuar la pregunta nuevamente.

```
GO TO 42
```

Habr  resumen por impresora. Se define nombres de archivos de 'default'

```

50 IF (OPCION.EQ.3) ARCH = 'RESUMEN.GEN'
IF (OPCION.EQ.4) ARCH = 'RESUMEN.SHU'
IF (OPCION.EQ.5) ARCH = 'RESUMEN.NOD'
IF (OPCION.EQ.6.AND.NISLAS.EQ.1) ARCH = 'RESUMEN.ISL'
IF (OPCION.EQ.6.AND.NISLAS.NE.1) ARCH = 'RESUMEN.NIS'
IF (OPCION.EQ.7.AND.NISLAS.NE.1) ARCH = 'RESUMEN.ISL'

```

Se lee nombre de archivo para resumen.

```

53 CALL CUADRO
WRITE (LTERM, 55) ESC
WRITE (LTERM, 56) ESC, ARCH, ESC
55 FORMAT (A, '[10;4H', '¿ NOMBRE DEL ARCHIVO DE RESUMEN ? ')
56 FORMAT (A, '[10;39H', A40, A, '[10;39H', $)
READ(LTERM, 57) NAME
57 FORMAT(A40)

```

Si se responde con tecla de 'entrada', se adopta el nombre definido por 'default'.

```

IF (NAME.EQ.' ') GO TO 60
ARCH = NAME

```

El canal para grabar el archivo conteniendo el resumen es el 8. Este podr  ser consultado posteriormente.

```
60 IWR = 8
```

En este ciclo se calculan totales de generaci n, carga, p rdidas y aportaciones de compensadores en derivaci n por cada isla.

```
371 DO 620 KISLA = 1, NISLAS
```

Se inicializan variables en donde ir almacenando la suma parcial de cada total en la isla actual.

```
PLS = 0.0
```

```

QLS = 0.0
TPG = 0.0
TQG = 0.0
TPC = 0.0
TQC = 0.0
G   = 0.0

```

Se definen los límites de la lista de localidades contiguas de memoria localizada en ISLA(.), donde se encuentran los nodos de la isla actual.

```

      INICIO = INISLA(KISLA)
      IF (INICIO.LT.0) GO TO 620
      IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
      IF (KISLA.EQ.NISLAS) IFIN = NNODOS
      IF (NISLAS.EQ.1) GO TO 384
380  FORMAT(/20X, '*****',
      *      /20X, 'RESUMEN DE LA ISLA No. ', I2,
      *      /20X, '*****', /)
382  FORMAT(A, '[2;20H', '*****', /,
      *      A, '[3;20H', 'RESUMEN DE LA ISLA No. ', I2, /,
      *      A, '[4;20H', '*****')

```

Se inicia el ciclo para cada isla.

```

384      DO 500 JS = INICIO, IFIN
          I = ISLA(JS)

```

Se verifica que el nodo I esté conectado en esta isla. Si es mayor de cero, estará conectado.

```

      IF (I.LT.0) GO TO 500

```

Se suma la generación y carga a los totales correspondientes.

```

      TPG = TPG + PG(I)
      TQG = TQG + QG(I)
      TPC = TPC + PC(I)
      TQC = TQC + QC(I)

```

Se calcula magnitud de voltaje y el cuadrado de ésta, en base a las partes real e imaginaria del voltaje del nodo I.

```

      V1 = VRE(I)
      V2 = VIM(I)
      VI = V1 * V1 + V2 * V2
      VT = SQRT(VI)

```

Se guarda magnitud de voltaje y se calcula ángulo de fase en grados.

```

      VOLT(I) = VT
      IF (V1.NE.0.0) ANG = ATAN (V2 / V1)
      IF (V2.EQ.0.0) ANG = 0.0
      IF (V1.LT.0.0.AND.V2.LT.0.0) ANG = ANG - 3.141592654
      IF (V1.GT.0.0.AND.V2.GT.0.0)
      *
          ANG = ANG - (3.141592654*2.0)
      IF (V1.LT.0.0.AND.V2.GT.0.0) ANG = ANG - 3.141592654

```

```

IF (V1.GT.0.0.AND.V2.LT.0.0) ANG = ANG
ANG = ANG * 57.29577951
ANGULO(I) = ANG

```

Se verifica que se tiene compensación en derivación en el nodo actual.
Si NLT(I) = 0, no existe compensación en derivación en el nodo I.

```
IF(NLT(I).EQ.0) GO TO 440
```

Se busca en la información de compensadores en derivación si existe el nodo I. En este caso, se calcula su aportación al sistema en MVars.

```

QM = 0.0
IS = 0
436   IS = IS + 1
      IF(IS.GT.NSHNTS)GO TO 2637
      IF(NSHUNT(IS).NE.I) GO TO 436
      QM = - VI * SHUNT(IS)
      G = G - QM

```

Se calculan flujos de potencia desde el nodo I a nodos con los que está interconectado, los cuales se identifican en la lista de encadenamiento IFINPR(.) para circuitos π de elementos de transmisión.

```

440   KC = INICPR(I)
451   IF (KC.EQ.0) GO TO 500
      J = IPOSPR(KC)
      IF (J.LT.0) GO TO 480
      IF (NE(J).LT.0) GO TO 480
      IC = NR(J)
      YPQ2 = YPQ2E(J)
      GPQ2 = GPQ2E(J)
      IF(I.EQ.NE(J)) GO TO 454
      IC = NE(J)
      YPQ2 = YPQ2R(J)
      GPQ2 = GPQ2R(J)
454   V3 = VRE(IC)
      V4 = VIM(IC)
      A = V1 * V3 + V2 * V4
      B = V2 * V3 - V1 * V4
      GPQ = R1(J)
      BPQ = X1(J)
      PIJ = (VI*(GPQ+GPQ2)-GPQ*A- B*BPQ)*SBASE
      QIJ = (-VI*(YPQ2+BPQ)+A*BPQ-B*GPQ)*SBASE
      PLS = PLS + PIJ
      QLS = QLS + QIJ
      KC = IFINPR(KC)
      GO TO 451
500   CONTINUE

```

Se guardan los totales de generación, carga, pérdidas y aportaciones de compensadores en derivación para la isla actual.

```

TOTPG(KISLA) = TPG * SBASE
TOTQG(KISLA) = TQG * SBASE

```



```
TOTPC(KISLA) = TPC * SBASE
TOTQC(KISLA) = TQC * SBASE
PL(KISLA)     = PLS
QL(KISLA)     = QLS
ACDER(KISLA) = G * SBASE
```

620 CONTINUE

Se asigna a la variable IFLAG el valor de cero para indicar que no se ha abierto archivo para resumen por impresora.

```
IFLAG = 0
IF (OPCION.EQ.3) GO TO 700
IF (OPCION.EQ.4) GO TO 900
IF (OPCION.EQ.5) GO TO 1096
IF (OPCION.EQ.6.AND.NISLAS.EQ.1) GO TO 1500
IF (OPCION.EQ.6.AND.NISLAS.NE.1) GO TO 1050
IF (OPCION.EQ.7.AND.NISLAS.NE.1) GO TO 1050
```

C

C REPORTE DE GENERADORES

C

Se verifica que el reporte no es completo y no es de todos los nodos del sistema y que el dispositivo de salida no es la impresora.

700 IF (OPCION.NE.8.AND.OPCION.NE.6.AND.IWR.NE.8) GO TO 713

El resumen es por impresora y se verifica si fué abierto anteriormente el archivo correspondiente. Si no es así, aquí se abre.

IF (IFLAG.EQ.0) OPEN (UNIT = 8, FILE = ARCH)

IFLAG = 1 para indicar que el archivo para resumen ha sido abierto.

IFLAG = 1

Se imprime el encabezado para datos de nodos de voltaje controlado.

```
WRITE (IWR, 710)
710 FORMAT(/3X, 'N O D O      VOLTAJE  NODAL  ',
*          'POT. GENERACION LIMITES-QG(MVAr) ISLA',/,
*          'No. NOMBRE      MAG(pu)  GRADOS  ',
*          ' (MW) (MVAr)  INF.      SUP.  ')
GO TO 720
```

Imprime por pantalla encabezado de datos resumidos de nodos generadores

```
713 CALL CUADRO
ILE = 9
WRITE (*, 715) ESC, ESC
WRITE (*, 717) ESC, ESC, ESC, ESC
715 FORMAT(A, '[3;23H', 'RESUMEN DE NODOS DE GENERACION',/,
*          A, '[4;23H', '-----')
717 FORMAT(A, '[6;4H', 'N O D O      VOLTAJE  NODAL  ',/,
*          A, '[6;34H', 'POT. GENERACION LIMITES-QG(MVAr) ISLA',/,
*          A, '[7;3H', 'No. NOMBRE      MAG(pu)  GRADOS  ',/,
*          A, '[7;34H', ' (MW) (MVAr)  INF.      SUP.  ')

```

Se imprime información resumida de nodos de voltaje controlado por isla

```

720 DO 820 KISLA = 1, NISLAS
      INICIO = INISLA(KISLA)
      IF (INICIO.LT.0) GO TO 820
      IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
      IF (KISLA.EQ.NISLAS) IFIN = NNODOS
      DO 800 JS = INICIO, IFIN
        I = ISLA(JS)
        IF (I.LT.0) GO TO 800
        DO 730 J = 1, NNGEN
          IF (NPV(J).EQ.I) GO TO 740
730      CONTINUE
        GO TO 800
740      NOMB1 = DNOMB(I)
        VT = VOLT(I)
        ANG = ANGULO(I)
        A = PG(I) * SBASE
        B = QG(I) * SBASE
        C = QGMIN(J) * SBASE
        D = QGMAX(J) * SBASE
        IF (IWR.NE.8) GO TO 780
        WRITE (IWR,750) I, NOMB1, VT, ANG, A, B, C, D, KISLA
750      FORMAT (I3, 1X, A8, 4X, F6.4, 1X, F7.2, 1X,
              *                               4(F7.1, 2X), 2X, I2)
        GO TO 800
780      IF (ILE.LE.21) GO TO 785

```

Se termina una página en la pantalla y se prepara la siguiente.

```

      CALL ENCA
      CALL CUADRO
      ILE = 6
      WRITE (*, 787) ESC, ESC, ESC, ESC
787      FORMAT(A, '[3;4H',
              *           'N O D O      VOLTAJE  NODAL  ', '/',
              *           A, '[3;34H',
              *           'POT. GENERACION LIMITES-QG(MVAr) ISLA', '/',
              *           A, '[4;3H',
              *           'No.  NOMBRE  MAG(pu)  GRADOS  ', '/',
              *           A, '[4;34H',
              *           ' (MW)  (MVAr)  INF.    SUP.  ')
785      IF (ILE.LE.9) WRITE (LTERM,790) ESC, ILE, I, NOMB1,
              *           VT, ANG, ESC, ILE, A, B, C, D, KISLA
              IF (ILE.GT.9) WRITE (LTERM,793) ESC, ILE, I, NOMB1,
              *           VT, ANG, ESC, ILE, A, B, C, D, KISLA
790      FORMAT (A, '[' , I1, ' ; 3H', I3, 1X, A8, 2X, F6.4, 1X, F7.2,
              *           /, A, '[' , I1, ' ; 32H', 4(F7.1, 2X), 1X, I2)
793      FORMAT (A, '[' , I2, ' ; 3H', I3, 1X, A8, 2X, F6.4, 1X, F7.2,
              *           /, A, '[' , I2, ' ; 32H', 4(F7.1, 2X), 1X, I2)
      ILE = ILE + 1
800      CONTINUE
820      CONTINUE

```

Termina impresión del reporte resumido de nodos PV. Ahora, se revisa si se trata de un reporte de este tipo o uno completo (todas las opciones)

```
IF (OPCION.NE.3) GO TO 900
```

Se trata de un resumen de generadores, por lo que se cierra el archivo correspondiente, y se reinicializa el valor de IFLAG en cero.

```
CLOSE (UNIT = 8)
IFLAG = 0
CALL ENCA
```

Se regresa al menú de reportes resumidos.

```
GO TO 29
```

C

C RESUMEN DE COMPENSADORES EN DERIVACION DEL SISTEMA ELECTRICO.

C

```
900 CONTINUE
```

Se verifica que exista compensación en derivación en el sistema. En caso de no existir, se regresa a menú de reportes resumidos.

```
IF (NSHNTS.EQ.0) GO TO 29
```

Se verifica si el dispositivo de salida es la impresora. Si IWR = 8, entonces se verifica que el archivo no se ha abierto. En este caso se abre y se marca su apertura asignando a IFLAG el valor de 1.

```
IF (IWR.NE.8) GO TO 910
IF (IFLAG.EQ.0) OPEN (UNIT = 8, FILE = ARCH)
IFLAG = 1
```

Se imprime encabezado para esta opción.

```
WRITE (IWR, 905)
905 FORMAT(//11X,
* 'NODO CON SHUNT   SUSCEPT.(pu)   Q-GENERADA (Mvars)   ISLA')
GO TO 915
910 CALL CUADRO
WRITE (LTERM, 913) ESC
913 FORMAT(A, '[4;11H',
* 'NODO CON SHUNT   SUSCEPT.(pu)   Q-GENERADA (Mvars)   ISLA')
ILE = 6
```

Se define la isla actual para buscar información de compensadores.

```
915 DO 999 KISLA = 1, NISLAS
      INICIO = INISLA(KISLA)
      IF (INICIO.LT.0) GO TO 999
      IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
      IF (KISLA.EQ.NISLAS) IFIN = NNODOS
```

Una vez definida la isla actual, para cada nodo de la misma se busca información de compensación en derivación.

```
DO 960 JS = INICIO, IFIN
  I = ISLA(JS)
```

Si $I < 0$, el nodo ha sido desconectado de la isla actual.
Si $NLT(I) = 0$, el nodo actual (I) no tiene compensación conectada.

```
IF (I.LT.0) GO TO 960
IF(NLT(I).EQ.0) GO TO 960
```

Se busca el compensador en derivación conectado al nodo I.

```
DO 920 J = 1, NSHNTS
920      IF (NSHUNT(J).EQ.I) GO TO 925
GO TO 960
```

Se ha encontrado el compensador conectado al nodo I, y se realizan los cálculos de su aportación de potencia reactiva a la isla actual.

```
925      NOMB1 = DNOMB(I)
        VT = VOLT(I)
        QM = VT * VT * SHUNT(J) * SBASE
        IF (IWR.EQ.8) GO TO 950
```

Se imprime la aportación en pantalla. Primeramente se verifica si no es necesario un cambio de página ($ILE > 21$).

```
IF (ILE .LE. 21) GO TO 930
```

Se hace cambio de página.

```
ILE = 6
CALL ENCA
CALL CUADRO
WRITE (LTERM,913) ESC
```

Se imprime la información del nodo con compensación por pantalla.

```
930      IF (ILE.LE.9) WRITE (LTERM,935) ESC, ILE, DNOMB(I),
*          SHUNT(J), QM, KISLA
        IF (ILE.GT.9) WRITE (LTERM,937) ESC, ILE, DNOMB(I),
*          SHUNT(J), QM, KISLA
935      FORMAT(A, '[', I1, ';14H', A8, 7X, F8.5, 10X, F8.2, 11X, I2)
937      FORMAT(A, '[', I2, ';14H', A8, 7X, F8.5, 10X, F8.2, 11X, I2)
        ILE = ILE + 1
GO TO 960
```

Se imprime la información del nodo con compensación por impresora.

```
950      WRITE (IWR,955) DNOMB(I), SHUNT(J), QM, KISLA
955      FORMAT (14X, A8, 7X, F8.5, 10X, F8.2, 11X, I2)
960      CONTINUE
999 CONTINUE
```

Se verifica que se trata de un resumen de compensadores solamente. Esto ocurre cuando $OPCION = 4$.

```
IF (OPCION.NE.4) GO TO 1096
```

Se cierra el archivo del resumen de compensadores. Se reinicializa a IFLAG con el valor de cero. Se regresa a menú de reportes resumidos.

```
CLOSE (UNIT = 8)
CALL ENCA
IFLAG = 0
GO TO 29
```

C

C RESUMEN DE NODOS PARA TODO EL SISTEMA O POR ISLA.

C

Se imprime resumen de nodos conteniendo información de voltajes y carga nodales por isla. Para ésto se lee el número de la isla deseada. En caso de que exista sólo una isla eléctrica, esta parte del programa se deshabilita.

```
1050 CALL CUADRO
1055 WRITE (*,1069) ESC
      WRITE (LTERM,1072) ESC, NISLAS, ESC
1069 FORMAT (A, '[22;4H', '
*
1072 FORMAT (A, '[22;6H', '¿ CUAL ISLA [ num. entero entre 1 y ', I2,
* ' "entrada" para salir ] ? ', /, A, '[22;73H', '$)
      READ (LTERM, 1081) KISLA
1081 FORMAT (I2)
      JS = KISLA
```

Se verifica que el número de isla proporcionado esté correcto. Si el número es 0, se regresa al menú de reportes resumidos.

```
IF (JS.EQ.0) GO TO 29
IF (JS.GT.NISLAS) WRITE (1083) ESC, KISLA
IF (JS.GT.NISLAS) GO TO 1055
IF (INISLA(JS).GT.0) GO TO 1095
WRITE (*, 1085) ESC, JS
1083 FORMAT (A, '[10;18H', 'LA ISLA ', I2, ' NO EXISTE ')
1085 FORMAT (A, '[10;18H', 'LA ISLA ', I2, ' ESTA DESACTIVADA')
```

Se regresa a preguntar por el número de isla nuevamente.

```
GO TO 1055
```

Se trata de un reporte por isla.

```
1095 INIC1 = KISLA
      IFIN1 = KISLA
```

Si OPCION = 7 se trata de un reporte de totales.

```
IF (OPCION.EQ.7) GO TO 1600
GO TO 1100
```

Se trata de un reporte completo.

```
1096 INIC1 = 1
      IFIN1 = NISLAS
```

1100 CONTINUE

Se verifica si la salida es la impresora. Si IWR=8, entonces se revisa si el archivo está cerrado. En este caso se abre e IFLAG=1.

```
IF (IWR.NE.8) GO TO 1110
IF (IFLAG.EQ.0) OPEN (UNIT = 8, FILE = ARCH)
IFLAG = 1
```

Se imprime el encabezado de esta opción.

```
WRITE (IWR,1105)
1105 FORMAT(//4X,'No. NODO MAG.VOLT.(pu) ANG.(grados) ',
*         ' P-CARGA(MW) Q-CARGA(MVAr) ISLA')
GO TO 1115
1110 CALL CUADRO
WRITE (LTERM,1113) ESC, ESC
1113 FORMAT(A,'[4;3H',' No. NODO MAG.VOLT.(pu) ANG.(grados)',
*         '/,A,'[4;45H',' P-CARGA(MW) Q-CARGA(MVAr) ISLA')
ILE = 6
```

Se imprime información de cada nodo por isla. En caso de que la opción sea reporte completo, se imprimen todos los nodos del sistema.

```
1115 DO 1199 KISLA = INIC1, IFIN1
      JS = INISLA(KISLA)
      INICIO = ABS(JS)
      IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
      IF (KISLA.EQ.NISLAS) IFIN = NNODOS
      DO 1160 JK = INICIO, IFIN
        I = ISLA(JK)
        IF (I.LT.0) GO TO 1160
        VT = VOLT(I)
        IF (JS.NE.INICIO) VT = 0.0
        ANG = ANGULO(I)
        A = PC(I) * SBASE
        B = QC(I) * SBASE
        IF (IWR.EQ.8) GO TO 1150
```

Reporte por pantalla. Se verifica que no se ha llenado la página actual

```
IF (ILE .LE. 21) GO TO 1130
ILE = 6
CALL ENCA
CALL CUADRO
WRITE (LTERM,1113) ESC, ESC
1130 IF (ILE.LE.9) WRITE (LTERM,1135) ESC, ILE, I,
*         DNOMB(I), VT, ANG, ESC, ILE, A, B, KISLA
      IF (ILE.GT.9) WRITE (LTERM,1137) ESC, ILE, I,
*         DNOMB(I), VT, ANG, ESC, ILE, A, B, KISLA
1135 FORMAT(A,'[', I1, ',';3H', I3,2X, A8,3X, F7.4,7X, F7.2,/,
*         A,'[', I1, ',';46H', F8.2,6X, F8.2,7X, I2)
1137 FORMAT(A,'[', I2, ',';3H', I3,2X, A8,3X, F7.4,7X, F7.2,/,
*         A,'[', I2, ',';46H', F8.2,6X, F8.2,7X, I2)
ILE = ILE + 1
```

GO TO 1160

Se imprime el reporte en archivo para impresora.

```
1150          WRITE (IWR, 1155) I, DNOMB(I), VT, ANG, A, B, KISLA
1155          FORMAT(3X, I3, 2X, A8, 3X, F7. 4, 7X, F7. 2, 6X, F8. 2, 5X, F8. 2, 7X, I2)
1160          CONTINUE
1199 CONTINUE
          CALL ENCA
```

Si OPCION = 6, se regresa a preguntar por otra isla.

```
IF (OPCION.EQ.6.AND.NISLAS.GT.1) GO TO 1050
```

Se trata de un reporte completo, y se reportan totales de cada isla.

```
IF (OPCION.EQ.7.AND.NISLAS.EQ.1) GO TO 1490
IF (OPCION.EQ.8.AND.NISLAS.NE.1) GO TO 1490
```

Se tiene solo una isla en el sistema, por lo que el archivo para impresora se cierra, y se regresa al menú de reportes resumidos.

```
CLOSE (UNIT = 8)
GO TO 29
```

```
C
C RESUMEN DE TOTALES POR ISLA
C
```

Reporte de totales para todas las islas.

```
1490 INIC1 = 1
      IFIN1 = NISLAS
      GO TO 1600
```

Reporte de totales para una isla.

```
1500 INIC1 = 1
      IFIN1 = 1
      GO TO 1600
1550 INIC1 = KISLA
      IFIN1 = KISLA
```

Se verifica si el dispositivo de salida es la impresora. Si IWR = 8, entonces se verifica que el archivo no se ha abierto. En este caso se abre y se marca su apertura asignando a IFLAG el valor de 1.

```
1600 IF (IFLAG.EQ.0) OPEN (UNIT = 8, FILE = ARCH)
      IFLAG = 1
```

Se imprimen los totales de cada isla.

```
DO 1700 KISLA = INIC1, IFIN1
      TPG = TOTPG(KISLA)
      TPC = TOTPC(KISLA)
      TQG = TOTQG(KISLA)
      TQC = TOTQC(KISLA)
      PLS = PL(KISLA)
```

```

        QLS = QL(KISLA)
        G   = ACDER(KISLA)
1605    IF (IWR.NE.8) GO TO 1620

```

Impresión de totales para impresora.

```

        WRITE (IWR,380) KISLA
        WRITE (IWR,1612) TPG, TQG, TPC, TQC, G, PLS, QLS
1612    FORMAT(25X,'TOTAL DE GENERACION :',
*        /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*        //25X,'TOTAL DE CARGA :',
*        /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*        //10X,'APORTACION DE COMPENSADORES',
*        ' EN DERIVACION : ',F7.1,' MEGAVARS',
*        //25X,'PERDIDAS EN EL SISTEMA :',
*        /10X,F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS')
        GO TO 1700

```

Reporte de totales por pantalla.

```

1620    CALL CUADRO
        WRITE (IWR,382) ESC, ESC, KISLA, ESC
        WRITE (IWR,1626)ESC,ESC,TPG,TQG,ESC,ESC,TPC,TQC,ESC,G,
*        ESC,ESC,PLS,QLS
1626    FORMAT(A,'[6;25H','TOTAL DE GENERACION :',
*        /,A,'[8;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*        /,A,'[11;25H','TOTAL DE CARGA :',
*        /,A,'[13;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS',
*        /,A,'[16;10H','APORTACION DE COMPENSADORES',
*        ' EN DERIVACION : ',F7.1,' MEGAVARS',
*        /,A,'[19;25H','PERDIDAS EN EL SISTEMA :',
*        /,A,'[21;10H',F12.1,' MEGAWATTS',10X,F12.1,' MEGAVARS')
        CALL ENCA

```

1700 CONTINUE

Se cierra archivo para impresora y se regresa a menú de resúmenes.

```

        CLOSE (UNIT = 8)
        GO TO 29

```

Se envía mensaje por errores en datos de compensadores en derivación.

```

2637 WRITE(6,2639)
2639 FORMAT(//2X,'ERROR EN DATOS DE COMPENSADORES !!!!')
        GO TO 2999
C*****
C*   REPORTE PARA EL PROGRAMA DE ESTABILIDAD.   *
C*****
2700 NN1 = NNODOS
        CALL CUADRO

```

Se verifica el número de islas activas en el sistema eléctrico.

```

        DO I = 1, NNODOS
            IF (ISLA(I).LT.0) NN1 = NN1 - 1
        END DO

```


C
C SE INCLUYE GRABACION DE ARCHIVO DE FLUJOS
C

Se define nombre de archivo de 'default'.

```
ARCH = 'ESTABILI.FLU'
```

Se lee el nombre del archivo de flujos para estabilidad.

```
WRITE (LTERM, 2705) ESC, YELFORE, ESC
WRITE (LTERM, 2710) ESC, WHITEFORE, ESC, ARCH, ESC
2705 FORMAT (A, A4, A, '[10;3H',
*           '¿ ARCHIVO DE FLUJOS PARA ESTABILIDAD ? ')
2710 FORMAT (A, A4, A, '[10;43H', A36, A, '[10;43H', $)
READ (LTERM, 2713) NAME
```

Si la respuesta es la tecla de 'entrada', se adopta nombre de 'default'

```
IF (NAME.NE.' ') ARCH = NAME
2713 FORMAT(A37)
```

Se envía mensaje de la grabación del archivo a pantalla.

```
WRITE (LTERM, 2715) ESC, YELFORE, ESC, ESC, WHITEFORE, ESC, ARCH
2715 FORMAT(A, A4, A, '[15;4H', '*** SE GRABA EL ARCHIVO DE FLUJOS : ', /,
*           A, A4, A, '[15;40H', A37)
```

Efecto de atraso de impresión en pantalla.

```
DO I = 1, 1000
    WRITE (*, 2718) ESC
2718    FORMAT(A, '[18;3H', ' ')
END DO
```

Se abre el archivo de flujos para estabilidad y se inicia impresión.

```
OPEN(UNIT=8, FILE=ARCH, STATUS='UNKNOWN', FORM = 'FORMATTED',
* ACCESS='SEQUENTIAL')
WRITE(8, 2720) NN1, NLINES, NNGEN, NTAPS, NSHNTS, NISLAS, LONG, SBASE
2720 FORMAT(2X, 7(15, 2X), F10.3)
```

Se imprimen datos nodales.

```
DO I = 1, NNODOS
    IF (ISLA(I).GE.0)
*    WRITE (8, 2721) DNOMB(I), PC(I), QC(I), VRE(I), VIM(I), INIC(I)
2721    FORMAT (2X, A8, 4(F10.5, 2X), I4)
END DO
NL = NLINES - (NNODOS - NN1)
IFIN1 = NL - NLINES + NLINES
NL = 0
```

Se imprimen datos de elementos de transmisión.

```
DO 2730 I = 1, NNODOS
    J = ISLA(I)
    IF (J.LT.0) GO TO 2730
```

```

      KC = INICPR(J)
2722  IF (KC.EQ.0) GO TO 2730
      J = KC
      IF (NL.EQ.IFIN1) GO TO 2733
      IF (IPOSPR(J).LT.0) GO TO 2725
      WRITE(8,2724) NE(J),NR(J),R(J),X(J),YPQ2E(J),YPQ2R(J)
2724  FORMAT (2X,2(I5,2X),4(F10.5,2X))
      NL = NL + 1
2725  KC = IFINPR(KC)
      GO TO 2722
2730 CONTINUE

```

Se imprimen datos de nodos de voltaje controlado.

```

2733 DO 2734 I = 1, NNGEN
      J = NPV(I)
      AUXA = QGMAX(I) * SBASE
      AUXI = QGMIN(I) * SBASE
      WRITE (8, 2731) J,PG(J),QG(J),AUXA,AUXI,TIPGEN(I)
2731  FORMAT (2X,I5,4(2X,F10.5),I1)
2734 CONTINUE

```

Se imprimen datos de transformadores con tap variable.

```

      IF (NTAPS.EQ.0) GO TO 2740
      DO 2736 I = 1, NTAPS
2736  WRITE (8, 2737) NETR(I),NRTR(I),TAP(I)
2737  FORMAT (2X,2(I5,2X),F10.5)

```

Se imprimen datos de compensadores en derivación.

```

2740 IF (NSHNTS.EQ.0) GO TO 2750
      DO 2741 I = 1, NSHNTS
      WRITE (8, 2742) NSHUNT(I),SHUNT(I)
2742  FORMAT (2X,I5,2X,F10.5)
2741 CONTINUE

```

Se imprime el número del nodo compensador de cada isla, la matriz de admitancias nodal y se cierra el archivo de impresión, regresando a menú principal de reportes.

```

2750 DO 2755 I = 1, NISLAS
      IF (INISLA(I).GE.0) WRITE (8, 2752) ISLACK(I)
2752  FORMAT (2X,I5)
2755 CONTINUE
      DO 2765 I = 1, LONG
2765  WRITE (8, 2762) ICOL(I),YBUSR(I),YBUSI(I),ISP(I)
2762  FORMAT (2X,I5,2X,2(F10.5,2X),I4)
      CLOSE ( UNIT = 8)
      RETURN
2999 ABORTA = 1

```

Se termina el subprograma RESREP.

```

3000 RETURN
      END

```

d) Subprograma TOPCAM

Aún cuando este subprograma permite modificar el archivo de datos del sistema eléctrico de potencia, se ha considerado como una rutina de salida, debido a que se utiliza normalmente después de un estudio de flujos.

Las actividades de este subprograma son:

Adición y remoción de componentes del sistema eléctrico de potencia (nodos y/o elementos de transmisión) así como cambios en sus parámetros.

Este subprograma es accesado mediante la instrucción:

```
CALL TOPCAM (YPQ2E, YPQ2R, R, X, NODOS, DNOMB, GPQ2E, GPQ2R, R1, X1, ABORTA,
*          PG, QG, NPV, VESP, QGMIN, QGMAX, NNGEN, TIPGEN, NSHNTS, NSHUNT,
*          SHUNT, PC, QC, NISLAS, ISLA, INISLA, SBASE, TIPO, NAU, LINES,
*          FPCISLA, FQCISLA, NODFUE, KFUERA, LETR1, LETR2)
```

Una descripción de las variables del argumento y los bloques comunes involucradas en este subprograma se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NODOS	E/S	ENTERA	Número total de nodos en el sistema. Inicialmente, esta variable toma el valor de NNODOS. Sin embargo, puede modificarse por adición o remoción de nodos.
PG(NODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema.
QG(NODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.
PC(NODOS)	E/S	REAL	Potencia real de carga en cada nodo del sistema.
QC(NODOS)	E/S	REAL	Potencia reactiva de carga en cada nodo del sistema.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
TIPO(NODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
GPQ2E(NLINES)	E/S	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de envío.
GPQ2R(NLINES)	E/S	REAL	Valor de la conductancia correspondiente a la mitad del efecto en derivación de un elemento de transmisión, relacionado con el nodo de recepción.
R1(NLINES)	E/S	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	E/S	REAL	Susceptancia inductiva serie del circuito π en elementos de transmisión.
YPQ2E(NLINES)	E/S	REAL	Mitad del efecto capacitivo asignado al nodo de envío en el circuito π de cada elemento de transmisión.
YPQ2R(NLINES)	E/S	REAL	Mitad del efecto capacitivo asignado al nodo de recepción en el circuito π de cada elemento de transmisión.
SBASE	E	REAL	Potencia base del sistema eléctrico en MVA.
NSHNTS	E/S	ENTERA	Total de compensadores en derivación en el sistema eléctrico de potencia.
NSHUNT(NSHNTS)	E/S	ENTERA	Indica el nodo en donde se conecta un compensador en derivación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
SHUNT(NSHNTS)	E/S	REAL	Valor de la susceptancia del compensador en derivación conectado al nodo indicado en NSHUNT(.).
DNOMB(NODOS)	E/S	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema eléctrico.
INISLA(NISLAS)	E	ENTERA	Indica el número de nodo en donde se inicia la información topológica de la isla indicada por la posición de INISLA(.).
ISLA(NODOS)	E	ENTERA	Contiene los nodos del sistema, formando una lista encadenada para cada isla.
NNGEN	E/S	ENTERA	Total de nodos generadores en el sistema eléctrico.
NAU(NNGEN)	E/S	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
NPV(NNGEN)	E/S	ENTERA	Contiene en cada posición el número correspondiente a un nodo de generación.
QGMIN(NNGEN)	E/S	REAL	Límite mínimo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).
QGMAX(NNGEN)	E/S	REAL	Límite máximo de potencia reactiva generada en el nodo indicado en la misma posición en NPV(.).
TIPGEN(NNGEN)	E/S	ENTERA	Tipo de nodo de voltaje controlado. Si TIPGEN(k)=0, el nodo k es una máquina síncrona. Si TIPGEN(k) = 1, el nodo k es un compensador estático de VARs.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ABORTA	E/S	ENTERA	Indica si al salir de este subprograma se ejecuta la instrucción siguiente del programa principal (ABORTA = 0) ó se termina la sesión de simulación (ABORTA = 1).
LETR1	E/S	CHARACTER	Primer renglón descriptivo del archivo de datos del sistema eléctrico.
LETR2	E/S	CHARACTER	Segundo renglón descriptivo del archivo de datos del sistema eléctrico.
KFUERA	E/S	ENTERA	Total de nodos que han sido desconectados del sistema en sesiones anteriores del subprograma TOPCAM.
NODFUE(KFUERA)	E/S	ENTERA	Indica el número de nodo desconectado del sistema en sesiones anteriores del subprograma TOPCAM.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica donde se inicia la información referente a la posición de circuitos π , la cual se encuentra en IPOSPR(), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición INICPR(i).
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k) indica donde se encuentra un circuito π relacionado con el nodo i, donde k está definida por IFINPR(.).
NE(NLINES)	E/S	ENTERA	NE(i) indica el nodo de envío correspondiente al circuito π i-ésimo.
LINES	E/S	ENTERA	Número de elementos de transmisión. Inicialmente toma el valor de NLINES.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NR(NLINES)	E/S	ENTERA	NR(i) indica el nodo de recepción correspondiente al circuito π i-ésimo.
FPCISLA(NISLAS)	E/S	REAL	Factor de escalamiento para modificar la potencia real de carga en todos los nodos de una isla.
FQCISLA(NISLAS)	E/S	REAL	Factor de escalamiento para modificar la potencia reactiva de carga en todos los nodos de una isla.

Existen algunas variables auxiliares utilizadas en el subprograma, las cuales se describen en cada parte del código que se presenta a continuación.

Inicio del subprograma TOPCAM.

```

SUBROUTINE TOPCAM (YPQ2E, YPQ2R, R, X, NODOS, DNOMB, GPQ2E, GPQ2R, R1,
*                X1, ABORTA, PG, QG, NPV, VESP, QGMIN, QGMAX, NNGEN, TIPGEN,
*                NSHNTS, NSHUNT, SHUNT, PC, QC, NISLAS, ISLA, INISLA, SBASE,
*                TIPO, NAU, LINES, FPCISLA, FQCISLA, NODFUE, KFUERA, LETR1, LETR2)

```

Declaración de bloques comunes.

```

COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR

```

Dimensionamiento y clasificación de variables.

```

INTEGER NE(1000), NR(1000), NETR(300), NRTR(300), ISLA(550), OPCION,
* NPV(150), INISLA(10), INICPR(550), IPOSPR(2000), NAU(150), ABORTA,
* TIPGEN(150), TIPO(550), NSHUNT(150), DEPEND(15), POSAUX(5),
* IFINPR(2000), NODFUE(15)

REAL X(1000), GPQ2E(1000), X1(1000), GPQ2R(1000), R1(1000), VESP(150),
* PG(550), QGMIN(150), FPCISLA(10), YPQ2E(1000), YPQ2R(1000), TAP(300),
* FQCISLA(10), QGMAX(150), R(1000), PC(550), QC(550), SHUNT(150), QG(550)

CHARACTER TIPVAL(15)*3, DNOMB(550)*8, VALCHA(15)*12, ARCH3*12
CHARACTER ARCH1*12, ARCH2*4, LETR1*70, LETR2*70, STRING*60, ESC*1
CHARACTER RESPUE*2, ANSWER*2, NAME1*8, NAME2*8, LISTOP(15)*50
CHARACTER WHITEFORE*4, YELFORE*4, ENCABE*50, ALTERN*50, LETR3*70
PARAMETER (YELFORE=' [33m', WHITEFORE=' [37m' )

```

LOGICAL IEX
REAL*8 VALNUM(15)

A ESC se le asigna el valor en código ASCII de la tecla 'esc', para - crear efectos de pantalla. LINES se inicializa con el valor de NLINES. MODIF indica si existen modificaciones a datos actuales del sistema de potencia (MODIF = 1) y entonces se tiene la opción de grabar un archivo nuevo que las incluya. KONT es un contador con varias funciones. POSAUX(.) es un arreglo auxiliar para almacenar posiciones de elementos de transmisión. NPCION es un indicador de que se está en la función de adición de nodos al sistema eléctrico (NPCION ≠ 0). KIND indica el tipo de función que se está realizando: remoción (KIND=0), adición (KIND=1), modificación (KIND=2).

```
ESC = CHAR(27)
LINES = NLINES
KONT = 1
MODIF = 0
15 DO 16 I = 1, 5
16     POSAUX(I) = 0
NPCION = 0
```

Se presenta el menú principal de modificación de datos, en el cual se pueden seleccionar cualquiera de las tres funciones ya señaladas.

```
CALL INIMEN
ENCABE = 'MENU PRINCIPAL DE MODIFICACION DE DATOS '
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE FLUJOS '
LISTOP(2) = ' TERMINAR EL PROGRAMA '
LISTOP(3) = ' MODIFICACION DE PARAMETROS '
LISTOP(4) = ' ADICION DE ELEMENTOS O NODOS '
LISTOP(5) = ' REMOCION DE ELEMENTOS O NODOS '
IF (MODIF.EQ.1)
*LISTOP(6) = ' GRABAR ARCHIVO DE DATOS CON MODIFICACIONES '
OPCION = 1
CALL MENU (OPCION)
IF (OPCION.EQ.1) GO TO 2220
IF (OPCION.EQ.2) GO TO 2200
IF (OPCION.EQ.3) GO TO 30
IF (OPCION.EQ.4) GO TO 20
IF (OPCION.EQ.5) GO TO 25
IF (OPCION.EQ.6) GO TO 2000
```

Menú para adición de elementos o nodos.

```
18 CALL INIMEN
20 STRING = 'ADICION DE '
ENCABE = 'MENU PRINCIPAL DE ADICION DE NODOS/ELEMENTOS'
LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE FLUJOS '
LISTOP(3) = ' ADICION DE LINEAS DE TRANSMISION '
LISTOP(4) = ' ADICION DE TRANSFORMADORES '
LISTOP(5) = ' ADICION DE ELEMENTOS EN DERIVACION '
LISTOP(6) = ' ADICION DE NODOS '
```


KIND = 1
GO TO 35

Menú para eliminar elementos o nodos.

```

25 CALL INIMEN
   STRING = 'REMOCION DE '
   ENCABE = 'MENU PRINCIPAL DE REMOCION DE NODOS/ELEMENTOS'
   LISTOP(1) = 'REGRESAR A MENU PRINCIPAL DE CAMBIOS '
   LISTOP(2) = 'REGRESAR A MENU PRINCIPAL DE FLUJOS '
   LISTOP(3) = 'REMOCION DE LINEAS DE TRANSMISION '
   LISTOP(4) = 'REMOCION DE TRANSFORMADORES '
   LISTOP(5) = 'REMOCION DE ELEMENTOS EN DERIVACION '
   LISTOP(6) = 'REMOCION DE NODOS '
   KIND = 0
   GO TO 35

```

Menú para modificar parámetros de elementos o nodos ya existentes.

```

30 CALL INIMEN
   STRING = 'MODIFICACION DE '
   ENCABE = 'MENU DE MODIFICACION DE PARAMETROS :'
   LISTOP(1) = 'REGRESAR A MENU PRINCIPAL DE CAMBIOS '
   LISTOP(2) = 'REGRESAR A MENU PRINCIPAL DE FLUJOS '
   LISTOP(3) = 'MODIFICAR LINEAS DE TRANSMISION '
   LISTOP(4) = 'MODIFICAR TRANSFORMADORES '
   LISTOP(5) = 'MODIFICAR ELEMENTOS EN DERIVACION '
   LISTOP(6) = 'MODIFICAR CARGAS '
   LISTOP(7) = 'MODIFICAR GENERADORES '
   KIND = 2
35 OPCION = 1
   MODIF = 1
   CALL MENU (OPCION)
   IF (KIND.EQ.2) GO TO 37
   IF (OPCION.EQ.1) GO TO 15
   IF (OPCION.EQ.2) GO TO 2220
   IF (OPCION.EQ.3) GO TO 100
   IF (OPCION.EQ.4) GO TO 200
   IF (OPCION.EQ.5) GO TO 400
   IF (OPCION.EQ.6) GO TO 500
   GO TO 35
37 MODIF = 1
   IF (OPCION.EQ.1) GO TO 15
   IF (OPCION.EQ.2) GO TO 2220
   IF (OPCION.EQ.3) GO TO 1100
   IF (OPCION.EQ.4) GO TO 1300
   IF (OPCION.EQ.5) GO TO 1450
   IF (OPCION.EQ.6) GO TO 1500
   IF (OPCION.EQ.7) GO TO 1700
   GO TO 35

```

C*****SE ADICIONAN, MODIFICAN O ELIMINAN LINEAS DE TRANSMISION

100 CALL INIMEN

VALCHA(3) y VALCHA(4) contendrán los nombres de los nodos de envío y recepción de la línea de transmisión que se adiciona, modifica o elimina.

```
VALCHA(3) = 'XXXXXXXX'
VALCHA(4) = VALCHA(3)
IF (KIND.NE.2) GO TO 102
```

Se define otra parte del menú para modificar líneas de transmisión.

```
ENCABE = 'MENU PARA MODIFICAR LINEAS DE TRANSMISION'
LISTOP(1) = ' REGRESAR A MENU DE MODIFICACION DE DATOS '
GO TO 106
102 IF (KIND.EQ.1) GO TO 103
```

Se define otra parte del menú para eliminar líneas de transmisión.

```
ENCABE = 'MENU PARA REMOVER LINEAS DE TRANSMISION'
LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE REMOCION '
GO TO 106
```

Se define otra parte del menú para adicionar líneas de transmisión. A TIPVAL(5), ..., TIPVAL(7) se les asigna 'REA' para indicar que las variables que manejará el menú en las opciones 5, 6, 7 serán reales, e inicialmente tendrán un valor de cero, definido en VALNUM(5), ..., VALNUM(7).

```
103 DO 104 I = 5, 7
      TIPVAL(I) = 'REA'
104   VALNUM(I) = 0.0
LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE ADICION '
LISTOP(5) = ' RESISTENCIA SERIE DE LA LINEA '
LISTOP(6) = ' REACTANCIA SERIE DE LA LINEA '
LISTOP(7) = ' MITAD DE LA ADMITANCIA EN DERIVACION '
ENCABE = ' MENU DE ADICION DE LINEAS DE TRANSMISION'
```

Estas opciones son comunes a las actividades de cambiar, adicionar o eliminar líneas de transmisión.

```
106 LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
LISTOP(3) = ' NOMBRE DEL NODO DE ENVIO '
LISTOP(4) = ' NOMBRE DEL NODO DE RECEPCION '
109 OPCION = 1
110 FORMAT (A8)
CALL MENU (OPCION)
NAME1 = VALCHA(3)
NAME2 = VALCHA(4)
```

Se verifican los nombres de nodos de envío y de recepción leídos. Si NAME1=NAME2, entonces no se leyó nombre alguno, indicando que se desea salir de esta opción, por lo que se regresa a menú principal de cambios

```
IF (NAME1.EQ.NAME2.AND.OPCION.EQ.2) GO TO 15
IF (KIND.EQ.0.AND.NAME1.EQ.NAME2.AND.OPCION.EQ.1) GO TO 25
IF (KIND.EQ.1.AND.NAME1.EQ.NAME2.AND.OPCION.EQ.1) GO TO 18
```

Se verifica que los nombres de nodos estén correctos. Si alguno no está incluido en el arreglo DNOMB(.), entonces se le asigna 'XXXXXXXX' a la variable correspondiente de VALCHA(.), y se regresa a leer nuevamente.

```

ITIP = 1
CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME2, IEX)
IF (JX.EQ.0) VALCHA(3) = 'XXXXXXXX'
IF (JY.EQ.0) VALCHA(4) = 'XXXXXXXX'
IF (.NOT.IEX) GO TO 109
120 FORMAT(A, '[15;5H', '** EL NODO ', A8, ' NO EXISTE ')

```

Los nombres de nodos están correctos y se transfiere el control de ejecución del programa según la función definida en el menú principal.

```

147 IF (KIND.NE.1) GO TO 160

```

Se agrega una línea de transmisión.

```

NN1 = LINES + 1
NE(NN1) = JX
NR(NN1) = JY
R (NN1)   = VALNUM(5)
X (NN1)   = VALNUM(6)
YPO2E(NN1) = VALNUM(7)
YPO2R(NN1) = VALNUM(7)
GPO2E(NN1) = 0.0
GPO2R(NN1) = 0.0

```

Si se ha leído valor de resistencia o reactancia distinto de cero, se da de alta la nueva línea y se incrementa el número total de las mismas en NLINES + 1.

```

IF(R(NN1).NE.0.0.OR.X(NN1).NE.0.0) LINES = NN1

```

Se hace INDL = 1 para indicar que se ha dado de alta una nueva línea entre dos nodos ya existentes antes de entrar en este subprograma.

```

IF(R(NN1).NE.0.0.OR.X(NN1).NE.0.0) INDL = 1

```

Si NPCION = 0, se trata de una nueva línea entre un nodo nuevo y uno ya existente.

```

IF (NPCION.NE.0) GO TO 700

```

Si OPCION=1, regresa a menú principal de adición de nodos o elementos.

```

IF (OPCION.EQ.1) GO TO 18

```

Si OPCION=2, regresa a menú principal de modificación de datos.

```

IF (OPCION.EQ.2) GO TO 15

```

Se modifica o elimina la línea de transmisión seleccionada.

```

160 KC1 = INICPR(JX)

```

Se verifica que la línea de transmisión no sea transformador.

```
DO I = 1, NTAPS
  IF (NETR(I).EQ.JX.AND.NRTR(I).EQ.JY) GO TO 193
  IF (NRTR(I).EQ.JX.AND.NETR(I).EQ.JY) GO TO 193
END DO
```

Se verifica que la línea de transmisión seleccionada no esté conectada en paralelo con otras líneas. Si ésto ocurre, entonces se mostrará cada línea de este conjunto para tener oportunidad de revisarlo.

```
163 KONT = 0
164 IF (KC1.EQ.0) GO TO 167
      I = IPOSPR(KC1)
      IF (I.LT.0) GO TO 166
      IF (NE(I).EQ.JX.AND.NR(I).EQ.JY) GO TO 165
      IF (NE(I).EQ.JY.AND.NR(I).EQ.JX) GO TO 165
      GO TO 166
165      KONT = KONT + 1
      POSAUX (KONT) = I
166      KC1 = IFINPR(KC1)
      GO TO 164
```

KONT = 1, indica que no existen líneas conectadas en paralelo.
KONT = 0, indica que no existe la línea de transmisión seleccionada.

```
167 IF (KONT.EQ.1) GO TO 190
      IF (KONT.EQ.0) GO TO 178
```

Existen líneas conectadas en paralelo y se muestra cada una de ellas al usuario para verificar su información.

```
DO 174 I = 1, KONT
  J = POSAUX(I)
168  CALL CUADRO
      ANSWER = 'NO'
      WRITE(*, 169)ESC, WHITEFORE, ESC, ESC, YELFORE, ESC, J, ESC, VALCHA(3),
*          ESC, VALCHA(4), ESC, R(J), ESC, X(J), ESC, YPQ2E(J), ESC
      IF(KIND.EQ.0) WRITE(*, 170)ESC, ESC, WHITEFORE, ESC, ANSWER, ESC
      IF(KIND.EQ.2) WRITE(*, 171)ESC, ESC, WHITEFORE, ESC, ANSWER, ESC
169  FORMAT(A, A4,
*      A, '[5;20H', 'LA LINEA DE TRANSMISION CON LOS DATOS:',
*      A, A4, '/', A, '[7;20H', 'LINEA                               No. : ', I3,
*      '/', A, '[9;20H', 'NODO      DE                               ENVIO : ', A8,
*      '/', A, '[11;20H', 'NODO      DE                               RECEPCION : ', A8,
*      '/', A, '[13;20H', 'RESISTENCIA                               SERIE : ', F8.5,
*      '/', A, '[15;20H', 'REACTANCIA                               SERIE : ', F8.5,
*      '/', A, '[17;20H', 'MITAD ADMITANCIA EN DERIVACION : ', F8.5,
*      '/', A, '[19;20H', 'ESTA EN PARALELO CON OTRA(S) LINEA(S)...')
170 FORMAT(A, '[21;20H', '¿ DESEA ELIMINAR ESTE ELEMENTO [SI, NO] ? ',
*      A, A4, A, '[21;64H', A2, A, '[21;64H', '$)
171 FORMAT(A, '[21;20H', '¿ DESEA MODIFICAR ESTE ELEMENTO [SI, NO] ? ',
*      A, A4, A, '[21;64H', A2, A, '[21;64H', '$)
      READ(*, 592) RESPUE
```

```

CALL SINO (RESPUE, ANSWER)
IF (RESPUE.EQ.'NO') GO TO 174
IF (RESPUE.EQ.'SI') GO TO 191
CALL CUADRO
GO TO 168
174 CONTINUE
178 CALL CUADRO
IF (KONT.NE.0) GO TO 100

```

Se imprime letrero de que la línea seleccionada no existe.

```

WRITE (*, 180) ESC, NAME1, NAME2
180 FORMAT(A, '[19;10H', 'EL ELEMENTO CON LOS NODOS ', A8,
*          ' Y ', A8, ' NO EXISTE...')
CALL ENCA
GO TO 109

190 J = POSAUX(1)
191 IF (KIND.EQ.2) GO TO 1235
IF (KIND.EQ.0) GO TO 196
193 CALL CUADRO

```

Se imprime letrero de que la línea seleccionada es un transformador.

```

WRITE (*, 194) ESC, NAME1, NAME2
194 FORMAT(A, '[19;10H', 'EL ELEMENTO CON LOS NODOS ', A8,
*          ' Y ', A8, ' ES TRANSFORMADOR...')
CALL ENCA
GO TO 100

```

Se imprime letrero de que la línea seleccionada es eliminada.

```

196 CALL CUADRO
WRITE(*, 169)ESC, WHITEFORE, ESC, ESC, YELFORE, ESC, J, ESC, VALCHA(3),
*          ESC, VALCHA(4), ESC, R(J), ESC, X(J), ESC, YPQZE(J), ESC
WRITE (*, 197) ESC, ESC, WHITEFORE, ESC
197 FORMAT (A, '[19;20H', '
*          A, A4, A, '[19;20H', 'ES REMOVIDA')

```

Se cambia el signo al número correspondiente al nodo de envío para que en procesos posteriores no pueda ser accesada la línea de transmisión, lo cual es equivalente a borrar su información del archivo de datos.

```
NE(J) = - NE(J)
```

Se regresa al menú principal de remoción de nodos o elementos.

```

CALL ENCA
GO TO 25
C***** SE ADICIONAN O ELIMINAN TRANSFORMADORES
200 CALL INIMEN

```

Se hace INDL=1 para indicar que un nuevo transformador, entre dos nodos ya existentes antes de entrar en este subprograma, ha sido dado de alta

```
INDL = 1
```

Se inicializa menú para adicionar o eliminar transformadores.

```

VALCHA(3) = 'XXXXXXXX'
VALCHA(4) = 'XXXXXXXX'
IF (KIND.EQ.1) GO TO 203
ENCABE = 'MENU PARA REMOVER TRANSFORMADORES '
LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE REMOCION '
GO TO 206

```

Variables reales y con valor de cero en las opciones 5, 6, 7.

```

203 DO 204 I = 5, 7
      TIPVAL(I) = 'REA'
204   VALNUM(I) = 0.0
      LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE ADICION '
      LISTOP(5) = ' RESISTENCIA SERIE DEL TRANSFORMADOR '
      LISTOP(6) = ' REACTANCIA SERIE DEL TRANSFORMADOR '
      LISTOP(7) = ' VALOR DEL TAP DEL TRANSFORMADOR '
      LISTOP(8) = ' CIRCUITO II DEL NUEVO TRANSFORMADOR '
      ENCABE = 'MENU DE ADICION DE NUEVOS TRANSFORMADORES '
206 LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
      LISTOP(3) = ' NOMBRE DEL NODO CONECTADO AL TAP '
      LISTOP(4) = ' NOMBRE DEL NODO A LADO OPUESTO DEL TAP '
209 OPCION = 1
      CALL MENU (OPCION)
      NAME1 = VALCHA(3)
      NAME2 = VALCHA(4)
      IF (NAME1.EQ.NAME2.AND.OPCION.EQ.1.AND.KIND.EQ.0) GO TO 25
      IF (NAME1.EQ.NAME2.AND.OPCION.EQ.2.AND.KIND.EQ.0) GO TO 15
      IF (NAME1.EQ.NAME2.AND.OPCION.EQ.1.AND.KIND.EQ.1) GO TO 18
      IF (NAME1.EQ.NAME2.AND.OPCION.EQ.2.AND.KIND.EQ.1) GO TO 15

```

Se verifica nombre de los nodos extremos del transformador.

```

ITIP = 1
CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME2, IEX)

```

Si el nombre del nodo del tap es incorrecto, se asigna a VALCHA(3) el valor de 'XXXXXXXX'. Lo mismo se hace con el nombre del nodo contrario al tap del transformador.

```

IF (JX.EQ.0) VALCHA(3) = 'XXXXXXXX'
IF (JY.EQ.0) VALCHA(4) = 'XXXXXXXX'
IF (.NOT.IEX) GO TO 209

```

Los dos nombres son correctos. Ahora se verifica el tipo de función que se desea efectuar con el transformador.

```

IF (KIND.EQ.1) GO TO 300

```

La función es de cambiar datos del transformador o eliminarlo. Aquí, se revisa el número de transformadores entre los nodos definidos arriba.

```

IND = 0
DO J = 1, NTAPS
  IF (JX.EQ.NETR(J).AND.JY.EQ.NRTR(J)) IND = IND + 1

```

END DO

Se revisa la información de cada transformador entre los nodos ya definidos. Si existen dos o más transformadores en paralelo (IND > 1), se muestra la información de cada uno de ellos y se pregunta si se desea eliminar.

```
KONT = 0
DO 270 J = 1, NTAPS
  T = TAP(J)
  IF (JX.NE.NETR(J).OR.JY.NE.NRTR(J)) GO TO 265
```

Se verifica si existe un transformador entre los nodos ya definidos.

```
IF (IND.EQ.1) GO TO 280
```

Se verifica si el transformador no ha sido ya eliminado.

```
IF (T.LT.0.0) GO TO 270
```

Existen transformadores en paralelo entre los dos nodos definidos y se procede a preguntar si se desea su eliminación.

```
261      CALL CUADRO
        T2 = ABS(T)
        RESPUE = 'NO'
        WRITE (*,262)ESC,YELFORE,ESC,ESC,J,ESC,NAME1,ESC,NAME2,
*          ESC,T2,ESC,ESC,ESC,WHITEFORE,ESC,RESPUE,ESC
262      FORMAT(A,A4,A,'[6;20H','EL TRANSFORMADOR CON LOS DATOS:',
*          /,A,'[8;20H','TRANSFORMADOR No. : ',I3,
*          /,A,'[10;20H','NODO LADO DEL TAP : ',A8,
*          /,A,'[12;20H','NODO LADO OPUESTO : ',A8,
*          /,A,'[14;20H','VALOR DEL TAP : ',F8.5,
*          /,A,'[16;20H','ESTA EN PARALELO CON OTRO(S) TRANSF.'
*          /,A,'[19;20H','¿ DESEA ELIMINAR ESTE ELEMENTO '
*          '[SI, NO] ? ',A,A4,A,'[19;64H',A2,A,'[19;64H',$(
        READ(*,592) RESPUE
        CALL SINO (RESPUE, 'NO')
        IF (RESPUE.EQ.'NO') GO TO 269
        IF (RESPUE.EQ.'SI') GO TO 280
        GO TO 261
265      IF (JX.NE.NRTR(J).OR.JY.EQ.NETR(J)) GO TO 270
```

Los nombres de los nodos extremos del transformador se proporcionaron en forma invertida.

```
        CALL CUADRO
        WRITE (*, 279) ESC, ESC, DNOMB(JY), DNOMB(JX)
        WRITE (*, 267) ESC
267      FORMAT (A,'[24;15H',
*          ,
*          CALL ENCA
        GO TO 200
```

Se cambia el signo del tap del transformador eliminado para que no sea accesado posteriormente, a menos que se aplique la función de adición de transformadores entre estos dos nodos ya definidos.

```

269          IF (T.GT.0.0) TAP(J) = - T
              KONT = KONT + 1
270 CONTINUE
          IF (IND.NE.0) GO TO 291
          CALL CUADRO
          WRITE (*, 180) ESC, NAME1, NAME2
          CALL ENCA
          IF (KIND.EQ.0) GO TO 200
          GO TO 203
279 FORMAT(A, '[19;10H', 'LOS NODOS DE ESTE TRANSF. ESTAN INVERTIDOS:',
* /, A, '[20;10H', 'NODO LADO TAP: ', A8, 3X, 'NODO CONTRARIO TAP: ', A8)

```

Se imprime letrero de que se elimina el transformador seleccionado.

```

280 CALL CUADRO
          WRITE (*, 282) ESC, ESC, J, ESC, NAME1, ESC, NAME2
282 FORMAT (A, '[10;15H', 'SE ELIMINA EL TRANSFORMADOR CON LOS DATOS:'
* /, A, '[12;20H', 'TRANSFORMADOR          No. : ', I3
* /, A, '[14;20H', 'NODO AL LADO DEL TAP : ', A8
* /, A, '[16;20H', 'NODO CONTRARIO AL TAP : ', A8)

```

Se actualiza la información topológica del transformador eliminado.

```

          IND = 0
          KC1 = INICPR(JX)
286 IF (KC1.EQ.0) GO TO 290
          I = IPOSPR(KC1)
          IF (I.LT.0) GO TO 288
          IF (NE(I).NE.JX.OR.NR(I).NE.JY) GO TO 288

```

Si KONT = IND, entonces, se ha identificado al transformador a eliminar

```

          IF (IND.NE.KONT) GO TO 287
          NE(I) = - NE(I)
          IND = 0
          KC1 = 0
          GO TO 290
287          IND = IND + 1
288          KC1 = IFINPR(KC1)
          GO TO 286
290 NETR(J) = - NETR(J)

```

Se regresan los taps a su valor original (mayor que cero).

```

291 DO 292 J = 1, NTAPS
292          TAP(J) = ABS (TAP(J))
          CALL ENCA
          GO TO 25
300 SH = VALNUM(7)

```


Se revisa que la resistencia o reactancia serie del transformador sean distintas de cero. Además, se verifica si se trata de un transformador entre nodos ya existentes o entre un nodo nuevo y uno ya existente.

```
IF (VALNUM(6).EQ.0.0.AND.SH.EQ.0.0.AND.NPCION.NE.0.
*
AND.INDL.EQ.0)GO TO 700
```

Tanto resistencia como reactancia serie son iguales a cero y la opción no es de proporcionar datos del circuito II del transformador.

```
IF (VALNUM(6).EQ.0.0.AND.SH.EQ.0.0.AND.OPCION.NE.8) GO TO 200
```

Se calcula el circuito II del nuevo transformador adicionado.

```
LINES = LINES + 1
NTAPS = NTAPS + 1
NETR(NTAPS) = JX
NRTR(NTAPS) = JY
NE(LINES) = JX
NR(LINES) = JY
R(LINES) = VALNUM(5)
X(LINES) = VALNUM(6)
TAP(NTAPS) = SH
YPQ2E(LINES) = 0.0
YPQ2R(LINES) = 0.0
IF (OPCION.EQ.8) GO TO 360
```

Se verifica que tanto el tap y la reactancia serie no sean cero.

```
IF (TAP(NTAPS).NE.0.0.AND.X(LINES).NE.0.0) GO TO 301
```

Tanto el tap como la reactancia serie son cero, por lo que no se da de alta el nuevo transformador.

```
LINES = LINES - 1
NTAPS = NTAPS - 1
```

Si se cumple esta condición el control se transfiere al menú de interconexiones de un nuevo nodo.

```
IF (INDL.EQ.0.AND.NPCION.NE.0) GO TO 700
GO TO 200
```

C***** SE CALCULA EL CIRCUITO II DEL NUEVO TRANSFORMADOR.

```
301 IF (TAP(NTAPS).LE.0.5) TAP(NTAPS) = 1.0
IND = 0
J = NTAPS
K = LINES
310 YPQ2R(K) = YPQ2E(K)
GPQ2E(K) = 0.0
GPQ2R(K) = 0.0
T = R(K) * R(K) + X(K) * X(K)
R1(K) = R(K) / T
X1(K) = - X(K) / T
T = ABS( TAP(J) )
YPQ2E(K) = X1(K) * (1.0 - T) / (T * T)
```

```

      YPQ2R(K) = X1(K) * (T - 1.0) / T
      R1(K) = R1(K) / T
      X1(K) = X1(K) / T
      TAP(J) = - T
      GPQ2E(K) = R1(K) * (1.0 - T) / (T * T)
      GPQ2R(K) = R1(K) * (T - 1.0) / T
350      IF (T.EQ.1.0) YPQ2E(K) = YPQ2R(K) - 0.00001
      IF (TAP(J).GT.0.0) TAP(J) = - TAP(J)

```

Se hace INDL=1 para indicar que un nuevo transformador, entre dos nodos ya existentes antes de entrar en este subprograma, ha sido dado de alta

```

      INDL = 1
      IF (IND.EQ.1388) GO TO 1388
      IF (KIND.EQ.2) GO TO 1300
      IF (NPCION.NE.0) GO TO 700
      IF (OPCION.EQ.1) GO TO 18
      IF (OPCION.EQ.2) GO TO 15
      GO TO 200

```

Se presenta el menú de modificación de parámetros del circuito Π del transformador de interés.

```

360 CALL INIMEN
      LISTOP(1) = ' REGRESAR A MENU PRINCIPAL DE ADICION '
      DO 366 I = 3, 8
          TIPVAL(I) = ' REA '
366      VALNUM(I) = 0.0
      I = LINES
      ENCABE = ' MENU DE DATOS DEL CIRCUITO  $\Pi$  DEL NUEVO TRANSF. '
370 LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
      LISTOP(3) = ' SUSCEPTANCIA EN DERIVACION LADO TAP '
      LISTOP(4) = ' CONDUCTANCIA EN DERIVACION LADO TAP '
      LISTOP(5) = ' SUSCEPTANCIA EN DERIVACION LADO OPUESTO '
      LISTOP(6) = ' CONDUCTANCIA EN DERIVACION LADO OPUESTO '
      LISTOP(7) = ' SUSCEPTANCIA SERIE DEL TRANSFORMADOR '
      LISTOP(8) = ' CONDUCTANCIA SERIE DEL TRANSFORMADOR '
      OPCION = 1

```

Se entra al subprograma de menús y al salir se revisa la susceptancia y la conductancia serie del transformador y determinar si los datos son aceptables. Además, si la opción no es cambiar datos, se regresa al menú de adición o eliminación de transformadores.

```

      CALL MENU (OPCION)
      IF (VALNUM(7).EQ.0.0.AND.VALNUM(8).EQ.0.0.AND.KIND.EQ.1) GO TO 200

```

Se actualizan los parámetros del circuito Π del transformador que se está modificando.

```

      YPQ2E(I) = VALNUM(3)
      GPQ2E(I) = VALNUM(4)
      YPQ2R(I) = VALNUM(5)
      GPQ2R(I) = VALNUM(6)
      X1(I) = VALNUM(7)

```

```
R1(I) = VALNUM(8)
```

Si la opción es modificar parámetros de transformadores ya existentes, se va a la etiqueta 1410.

```
IF (KIND.EQ.2) GO TO 1410
```

Se verifica que el nuevo transformador tenga el tap y la susceptancia serie distintas de cero, para darlo de alta definitivamente. Si esto no se cumple, NTAPS se reduce en 1 y el transformador no es dado de alta.

```
IF (X1(LINES).NE.0.0.AND.TAP(NTAPS).NE.0.0) GO TO 380
NTAPS = NTAPS - 1
GO TO 200
```

Se calcula la resistencia y la reactancia serie del transformador.

```
380 T = TAP(NTAPS)
X(LINES) = 1.0 / ( X1(LINES) * T )
R(LINES) = 0.0
IF (R1(LINES).EQ.0.0) GO TO 200
G = R1(LINES)
B = X1(LINES)
R(LINES) = T * G / ( G * G + B * B )
X(LINES) = - T * B / ( G * G + B * B )
INDL = 1
```

Si NPCION ≠ 0, se va al menú de interconexiones de un nuevo nodo. Si esto no ocurre y OPCION = 1, se regresa al menú principal de adiciones.

```
IF (NPCION.NE.0) GO TO 700
IF (OPCION.EQ.1) GO TO 18
```

Si cualquiera de las alternativas anteriores no se cumplen, se regresa al menú de adición o eliminación de transformadores.

```
GO TO 200
```

C***** SE ADICIONAN O ELIMINAN COMPENSADORES EN DERIVACION

Si la opción de cambios en datos de compensadores en derivación no es de remoción, se va al 450.

```
400 IF (KIND.NE.0) GO TO 450
```

Se verifica que haya compensación en derivación antes de intentar remover algún capacitor o reactor.

```
DO 405 I = 1, NSHNTS
405 IF (NSHUNT(I).GT.0) GO TO 450
CALL CUADRO
WRITE (*, 410) ESC, ESC
410 FORMAT (A, '[10;10H', 'OPCION NO VALIDA... ', '/', A, '[12;10H',
* 'NO EXISTEN COMPENSADORES EN DERIVACION !!!')
CALL ENCA
GO TO 25
```

Existe compensación en derivación o se agrega un compensador.

```
450 CALL INIMEN
      OPCION = 3
      VALCHA(3) = 'XXXXXXXXX'
455 NAME1 = 'XXXXXXXXX'
      TIPVAL(4) = 'REA'
      VALNUM(4) = 0.0
```

Se prepara menú de remoción de compensación en derivación.

```
ENCABE = 'MENU PARA REMOVER COMPENSACION EN DERIVACION'
LISTOP(1) = 'REGRESAR A MENU PRINCIPAL DE REMOCION'
LISTOP(2) = 'REGRESAR A MENU PRINCIPAL DE CAMBIOS'
LISTOP(3) = 'NOMBRE DEL NODO DEL COMPENSADOR'
IF (KIND.EQ.0) GO TO 456
```

Se prepara menú para adición de nuevos compensadores.

```
ENCABE = 'MENU DE NUEVOS COMPENSADORES EN DERIVACION'
LISTOP(4) = 'SUSCEPTANCIA DEL COMPENSADOR'
LISTOP(1) = 'REGRESAR A MENU PRINCIPAL DE ADICION'
456 OPCION = 1
      CALL MENU (OPCION)
```

No se leyó nombre de nodo alguno y se regresa a menú principal de cambios en datos del sistema eléctrico de potencia.

```
457 IF (NAME1.EQ.VALCHA(3).AND.OPCION.EQ.2) GO TO 15
```

No se leyó nombre de nodo alguno y se regresa a menú principal de eliminación de nodos o elementos.

```
IF (NAME1.EQ.VALCHA(3).AND.OPCION.EQ.1.AND.KIND.EQ.0) GO TO 25
```

No se leyó nombre de nodo alguno y se regresa a menú principal de adición de nodos o elementos.

```
IF (NAME1.EQ.VALCHA(3).AND.OPCION.EQ.1.AND.KIND.EQ.1) GO TO 18
```

Se ha leído el nombre de un nodo, y se verifica que esté incluido en la lista de nombres de nodos DNOMB(.).

```
NAME1 = VALCHA(3)
ITIP = 2
CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME1, IEX)
```

Si JX = 0, indica que el nombre de nodo leído no existe en DNOMB(.). Por lo tanto, se regresa a realizar otro intento.

```
IF (JX.EQ.0) GO TO 400
```

Se leyó el nombre de nodo correctamente y ahora se verifica que éste tenga compensación para poder efectuar una modificación o remoción del compensador, o que no la tenga y poder dar de alta uno nuevo.

```

DO 482 I = 1, NSHNTS
      IF (NSHUNT(I).NE.JX) GO TO 482
      IF (KIND.EQ.0) GO TO 480
      CALL CUADRO
      WRITE (*,471) ESC, NAME1
471   FORMAT (A,'[19;10H','EL NODO ',A8,
      *      ' TIENE COMPENSACION EN DERIVACION')
      CALL ENCA
      GO TO 450
480   CALL CUADRO
      WRITE (*,485) ESC, NAME1
485   FORMAT (A,'[19;10H',
      *      'SE ELIMINA EL COMPENSADOR EN DERIVACION DEL NODO ',A8)
      IF (NSHUNT(I).GT.0) NSHUNT(I) = - NSHUNT(I)
      CALL ENCA
      GO TO 457
482 CONTINUE
      IF (KIND.NE.0) GO TO 488

```

Se ha detectado que el nodo donde se desea eliminar la compensación no tiene compensador alguno conectado. Aquí, se imprime este mensaje.

```

CALL CUADRO
WRITE (*,1475) ESC, NAME1
CALL ENCA
GO TO 450

```

Se verifica que el valor del nuevo o del ya existente compensador no tenga valor de cero para poder darlo de alta.

```

488 IF (VALNUM(4).EQ.0.0) GO TO 457
      NSHNTS = NSHNTS + 1
      NSHUNT(NSHNTS) = JX
      SHUNT (NSHNTS) = VALNUM(4)
495 FORMAT (F14.9)

```

Si NPCION ≠ 0, se va al menú de interconexiones de un nuevo nodo. Si esto no ocurre, se regresa al menú de compensadores en derivación.

```

IF (NPCION.NE.0) GO TO 700
GO TO 457

```

C*****SE ADICIONAN O SUPRIMEN NODOS DEL SISTEMA

```

500 CALL CUADRO

```

Se imprime la variable STRING y el letrero del format 507, para indicar la función de adición o remoción de nodos en el sistema eléctrico.

```

WRITE (*, 507) ESC, ESC, YELFORE,STRING
507 FORMAT (A,'[6;25H',A,A4,A13,'NODOS EN EL SISTEMA  ')
      WRITE (*,1102) ESC, WHITEFORE, ESC
508 WRITE (*, 510) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE

```

El simulador pide en pantalla el nombre del nodo.

```

510 FORMAT(A,'[10;5H',A,A4,'¿ NOMBRE DEL NODO ',
      * '[ máximo 8 caracteres ] ? ',/,A,'[10;65H',A,A4,$)

```

```
READ (*,110) NAME1
```

Si el nombre proporcionado es la tecla de 'entrada' (NAME1 = ' '), se regresa a menú principal de cambios en datos del sistema.

```
IF (NAME1.EQ.' ') GO TO 35
```

Se ha leído el nombre de un nodo y se verifica el tipo de función que se realizará con el mismo. Si es adición, se va al 530. Si es remoción, se verifica que el nombre del nodo esté en la lista DNOMB(.).

```
IF (KIND.EQ.1) GO TO 530
ITIP = 2
CALL VERIF (NODOS,ITIP,JX,JY,KONT,DNOMB,NAME1,NAME1,IEX)
IF (JX.EQ.0) WRITE (*, 517) ESC
IF (JX.EQ.0) GO TO 500
```

El nombre del nodo es correcto y se removerá del archivo de datos del sistema. Para ésto, es necesario que se remueva toda la información que lo involucre (carga, generación, compensación, transformadores, líneas)

```
512 FORMAT (A,'[10;10H','SE ELIMINA DEL SISTEMA EL NODO ',A8)
517 FORMAT (A,'[10;5H',
```

C***** SE ELIMINA EL NODO LEIDO

```
DO 519 I = 1, NODOS
  IF (JX .NE. ABS(ISLA(I)) ) GO TO 519
```

Se verifica que el nodo a remover no haya sido removido anteriormente, y se da de alta en la lista de nodos eliminados NODFUE(.).

```
IF (ISLA(I).LT.0) GO TO 529
DO L = 1, KFUERA
  IF (NODFUE(L).EQ.JX) GO TO 529
END DO
ISLA(I) = - ISLA(I)
KFUERA = KFUERA + 1
NODFUE(KFUERA) = JX
CALL CUADRO
WRITE (*,512) ESC, NAME1
CALL ENCA
GO TO 520
```

```
519 CONTINUE
```

Se cambia el signo del nodo de envío de los elementos de transmisión que tengan como extremo al nodo que se elimina, para no considerarlos en procesos posteriores. Lo mismo se hace con la lista de nodos de generación, transformadores y compensación en derivación.

```
520 DO 522 I = 1, LINES
522 IF (NE(I).EQ.JX.OR.NR(I).EQ.JX.AND.NE(I).GT.0) NE(I)=-NE(I)
DO 524 I = 1, NNGEN
524 IF (NPV(I).EQ.JX.AND.NPV(I).GT.0) NPV(I) = - NPV(I)
```

```

DO 526 I = 1, NTAPS
526     IF(NETR(I).EQ. JX. OR. NRTR(I).EQ. JX. AND. NETR(I).GT. 0)
*           NETR(I) = - NETR(I)
DO 528 I = 1, NSHNTS
528     IF(NSHUNT(I).EQ. JX. AND. NSHUNT(I).GT. 0)NSHUNT(I)=-NSHUNT(I)

```

Se regresa a leer el nombre de otro nodo a eliminar o adicionar.

GO TO 500

Se imprime letrero de que el nodo de interés ya había sido eliminado anteriormente.

```

529 CALL CUADRO
WRITE (*,120) ESC, NAME1
CALL ENCA

```

Se regresa a leer el nombre de otro nodo a eliminar o adicionar.

GO TO 500

C*****SE ADICIONA EL NODO LEIDO AL SISTEMA ELECTRICO.

530 NN = NODOS

Se inicializa IND con -1 para indicar que el nodo agregado no existía anteriormente.

```

IND = - 1
RESPUE = 'NO'
DO 568 I = 1, NODOS

```

Se verifica que el nombre del nuevo nodo no está en la lista de nombres de nodos actual, esto es, en DNOMB(.).

IF (DNOMB(I).NE.NAME1) GO TO 568

Se ha encontrado al nombre de nodo en DNOMB(.), y ahora se verifica si el nodo está dado de alta o ha sido eliminado anteriormente.

```

DO 531 L = 1, LINES
IF (NE(L).GT.0) GO TO 531
IF (ABS(NE(L)).EQ. I. OR. NR(L).EQ. I) IND = 1
531 CONTINUE
CALL CUADRO
IF (IND.GE.0) GO TO 533

```

El nodo está dado de alta en el archivo de datos.

```

WRITE (*,532) ESC, NAME1, ESC
532 FORMAT(A,'[4;10H','EL NODO ',A8,' YA EXISTE EN EL SISTEMA...'
* ,/,A,'[6;10H','NO ES NECESARIO ACTUALIZAR SU INFORMACION')
CALL ENCA
GO TO 500

```

El nodo había sido eliminado anteriormente. Por lo tanto, se pregunta si se desea dar de alta nuevamente.

533 IF (IND.EQ.1) WRITE (*,534) ESC, NAME1

```

534 FORMAT(A,'[4;10H','* EL NODO ',A8,' YA EXISTIA EN EL SISTEMA...')
535     ANSWER = 'SI'
        WRITE(*,536)ESC,ESC,YELFORE,ESC,ESC,WHITEFORE,ANSWER,ESC
536     FORMAT(A,'[10;7H',A,A4,
*       ¿ SE ACTUALIZA INFORMACION DEL NODO [ SI, NO ] ? '
*       ,/,A,'[10;68H',A,A4,A2,' ',A,'[10;68H',$,)
        READ(*,592) RESPUE
        CALL SINO (RESPUE, ANSWER)
        IF (RESPUE.EQ.'SI') GO TO 538
        IF (RESPUE.EQ.'NO') GO TO 570
        GO TO 535

```

Se da de alta la información referente al nodo.

```

538     IND = 0
        WRITE(*,517) ESC
        WRITE(*,540) ESC, ESC, PC(I), ESC, QC(I)
540     FORMAT(A,'[6;20H','SE DAN DE ALTA SUS DATOS DE CARGA :',/,
*       A,'[8;7H','POTENCIA REAL : ',F8.3,' MW',/,
*       A,'[8;41H','POTENCIA REACTIVA : ',F8.3,' MVar',/)
        DO 543 L = 1, NNGEN
        IF( ABS(NPV(L)).NE.I) GO TO 543
        NPV(L) = ABS(NPV(L))
        IND = L
        STRING = 'COMPENSADOR ESTATICO'
        IF(TIPGEN(L).EQ.0) STRING = 'MAQUINA SINCRONA'
        WRITE(*,541)ESC,ESC,PG(I),ESC,VESP(L), ESC, QGMAX(L),
*       ESC,QGMAX(L),ESC,DNOMB(ABS(NAU(L))), ESC, STRING
541     FORMAT(A,'[10;10H','SE DAN DE ALTA SUS DATOS DE GENERACION :',/,
*       A,'[12;10H','POTENCIA REAL : ',F8.3,' MW',/,A,'[12;44H',
*       'VOLTAJE : ',F8.3,' pu-KV',/,A,'[13;6H','LIMITE MAXIMO Qg'
*       ', ' : ',F8.3,' MVar',/,A,'[13;44H','LIMITE MINIMO Qg : ',
*       F8.3,' MVar',/,A,'[14;6H','NODO ALTA TENSION : ',A8,/,
*       A,'[14;38H','TIPO DE GENERADOR : ',A20)
        QG(I) = 0.0
543     CONTINUE
        WRITE(*,545) ESC, ESC
545     FORMAT(A,'[16;15H',
*       'SE DAN DE ALTA SUS ELEMENTOS DE TRANSMISION:',/,A,'[17;7H',
*       'NODO-ENVIO NODO-RECEPCION R-SERIE X-SERIE Y-DERIV/2')
        K = 18
        DO 548 L = 1, LINES
        J = ABS(NE(L))
        IF (J.NE.I.AND.NR(L).NE.I) GO TO 548
        NE(L) = J
        IF (K.LE.23) GO TO 548
        CALL ENCA
        CALL CUADRO
        WRITE(*,534) ESC, NAME1
        WRITE(*,545) ESC, ESC
        K = 18
546     WRITE(*,547)ESC,K,DNOMB(J),DNOMB(NR(L)),R(L),X(L),YPQ2E(L)
547     FORMAT(A,'[',I2,';8H',A8,6X,A8,6X,F7.4,3X,F7.4,4X,F7.4)

```



```

      K = K + 1
548  CONTINUE
      CALL ENCA
      K = 0
      DO 549 L = 1, NTAPS
549      IF (ABS(NETR(L)).EQ.1.OR.NRTR(L).EQ.1) K = 1
      IF (K.EQ.0) GO TO 555
      WRITE (*,534) ESC, NAME1
      K = 10
      WRITE (*,550)ESC, ESC
550  FORMAT(A,'[6;15H',
*      'SE DAN DE ALTA LOS DATOS DE TRANSFORMADORES:',/,A,'[8;10H',
*      'NUMERO NODO-LADO-DEL-TAP NODO-CONTRARIO-AL-TAP TAP')
      DO 553 L = 1, NTAPS
      J = ABS(NETR(L))
      IF (J.NE.1.AND.NRTR(L).NE.1) GO TO 553
      WRITE(*,552)ESC,K,L, DNOMB(J), DNOMB(NRTR(L)), TAP(L)
552  FORMAT(A,'[',I2,',12H',I3,8X,A8,15X,A8,8X,F7.5)
      NETR(L) = J
      K = K + 1
553  CONTINUE
555  DO 560 L = 1, NSHNTS
      J = ABS(NSHUNT(L))
      IF (J.NE.1) GO TO 560
      IF (K.LT.20) GO TO 556
      CALL ENCA
      CALL CUADRO
      WRITE (*,534) ESC, NAME1
      K = 10
556  K = K + 1
      WRITE (*,550)ESC,K,ESC,K+1,ESC,K+2,L,SHUNT(L)
557  FORMAT(A,'[',I2,',15H','SE DA DE ALTA SU ',
*      'COMPENSADOR EN DERIVACION:',/,A,'[',I2,',15H',
*      'NUMERO VALOR DEL COMPENSADOR',
*      ',/,A,'[',I2,',17H',I3,22X,F7.4)
      NSHUNT(L) = J
560  CONTINUE
      GO TO 579
568  CONTINUE

```

Se incrementa en 1 el número de nodos en el sistema.
--

```

      NN = NN + 1
      DNOMB(NN) = NAME1
      I = NN
570  PC(I) = 0.0
      QC(I) = 0.0
      PG(I) = 0.0
575  QG(I) = 0.0

```

Se da de baja el nodo de la lista de nodos eliminados.
--

```

579 DO 580 K = 1, KFUERA
580  IF (NODFUE(K).EQ.1) NODFUE(K) = 0

```

K = 1
KK = 0

Se compacta la información de la lista de nodos eliminados.

```
DO 581 L = 1, KFUERA
      NODFUE(K) = NODFUE(L)
      IF (NODFUE(L).NE.0) K = K + 1
581   IF (NODFUE(L).EQ.0) KK = KK + 1
      KFUERA = KFUERA - KK
      IF (RESPUE.EQ.'SI') GO TO 18
      IF (IND.EQ.0) GO TO 595
      IF (IND.GT.0) GO TO 600
```

Aquí, se lee toda la información del nodo que se está dando de alta.

CALL CUADRO

Se especifica el tipo de nodo (2: voltaje controlado, 3: carga).

```
585 RESPUE = 'PQ'
      WRITE (*, 590) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE, RESPUE, ESC
590 FORMAT(A, '[10;9H', A, A4, ' ¿ TIPO DEL NODO [CARGA : PQ, GENE',
* 'RADOR: PV ] ? ', /, A, '[10;68H', A, A4, A2, ' ', A, '[10;68H', '$)
      READ (*, 592) RESPUE
592 FORMAT(A2)
      IF (RESPUE.EQ.'pq'.OR.RESPUE.EQ.'PQ'.OR.RESPUE.EQ.'Pq'.OR.
* RESPUE.EQ.'pQ'.OR.RESPUE.EQ.' ') GO TO 595
      IF (RESPUE.EQ.'PV'.OR.RESPUE.EQ.'pv'.OR.RESPUE.EQ.'pV'.OR.
* RESPUE.EQ.'Pv') GO TO 600
      GO TO 585
```

El nuevo nodo es de carga.

```
595 TIPO(I) = 3
      CALL INIMEN
      TIPVAL(2) = 'REA'
      TIPVAL(3) = 'REA'
      VALNUM(2) = PC(I)
      VALNUM(3) = QC(I)
597 ENCABE = ' MENU DE ADICION DE NUEVOS NODOS DE CARGA '
      ALTERN = ' ( CUAL OPCION DESEA [1'
      LISTOP(1) = ' IR A MENU DE INTERCONEXION DEL NUEVO NODO '
      LISTOP(2) = ' CARGA REAL DEL NUEVO NODO (MW) '
      LISTOP(3) = ' CARGA REACTIVA DEL NUEVO NODO (MVar) '
598 OPCION = 1
      CALL MENU (OPCION)
      PC(I) = VALNUM(2)
      QC(I) = VALNUM(3)
      PC(I) = PC(I) / SBASE
      QC(I) = QC(I) / SBASE
```

Se inicializa el nombre del nodo al cual estará conectado el nuevo nodo

NAME2 = 'xxxxxxxx'

```
GO TO 680
600 CALL INIMEN
```

El nuevo nodo es de voltaje controlado. El número de nodos generadores se incrementa en 1. En NPV(NNGEN) se guarda el número del nuevo nodo.

```
TIPO (I) = 2
IF (IND.GT.0) GO TO 602
NNGEN = NNGEN + 1
NPV(NNGEN) = I
```

Se asignan valores iniciales a la información del nuevo nodo de voltaje controlado, los cuales se modificarán en el subprograma MENU.

```
DO 601 L = 2, 4
601 VALNUM(L) = 0.0
VALNUM(5) = 1.0
L = NNGEN
GO TO 610
```

La información del nodo dado de alta ya existe y se asigna como valores iniciales a los datos a ser modificados en el subprograma MENU.

```
602 L = IND
VALNUM(2) = PG(I) * SBASE
VALNUM(3) = QGMAX(L) * SBASE
VALNUM(4) = QGMIN(L) * SBASE
VALNUM(5) = VESP(L)
610 DO 612 J = 2, 5
612 TIPVAL(J) = 'REA'
VALCHA(6) = 'M-S'
VALCHA(7) = NAME1
ENCABE = 'MENU DE ADICION DE NUEVOS NODOS GENERADORES'
ALTERN = ' ( CUAL OPCION DESEA [1'
LISTOP(1) = ' IR A MENU DE INTERCONEXION DEL NUEVO NODO '
LISTOP(2) = ' POTENCIA REAL DE GENERACION (MW) '
LISTOP(3) = ' LIMITE MAXIMO DE POTENCIA REACTIVA (MVAr) '
LISTOP(4) = ' LIMITE MINIMO DE POTENCIA REACTIVA (MVAr) '
LISTOP(5) = ' VOLTAJE DE GENERACION (p.u.) '
LISTOP(6) = ' TIPO DE NODO DE GENERACION [CEV, M-S] '
LISTOP(7) = ' NODO DE ALTA TENSION (Máx. 8 caracteres) '
615 OPCION = 1
CALL MENU (OPCION)
```

Se asignan valores a las distintas variables del nuevo nodo P-V, y se verifica el tipo de generador y el nombre del nodo de alta tensión al estará conectado.

```
PG (I) = VALNUM(2)
QGMAX(L) = VALNUM(3) / SBASE
QGMIN(L) = VALNUM(4) / SBASE
VESP (L) = VALNUM(5)
IF (VALCHA(6).EQ.'M-S') TIPGEN(L) = 0
IF (VALCHA(6).EQ.'CEV') TIPGEN(L) = 1
IF (VALCHA(6).EQ.'CEV'.OR.VALCHA(6).EQ.'M-S') GO TO 617
```

```

VALCHA(6) = 'M-S'
GO TO 615
617 DO 620 M = 1, NN
      IF (DNOMB(M).NE.VALCHA(7)) GO TO 620
      IF (VALCHA(6).EQ.'CEV') NAU(L) = - M
      IF (VALCHA(6).EQ.'M-S') NAU(L) =  M
      GO TO 630
620 CONTINUE

```

Se ha encontrado que el nombre del nodo de alta tensión asociado es incorrecto, se imprime el mensaje correspondiente y se regresa a leer tal nombre nuevamente.

```

CALL CUADRO
WRITE (*, 120) ESC, VALCHA(7)
CALL ENCA
GO TO 610

```

Se ha definido correctamente la información del nuevo nodo y ahora se pasa a definir su forma de conectarse al sistema eléctrico.

```

630 NAME2 = 'xxxxxxxx'
      IF (NAU(L).GT.0) GO TO 670
      PG(I) = 0.0
      NAME2 = VALCHA(7)
      IF (ABS(NAU(L)).NE.1) GO TO 670
      VALCHA(7) = 'xxxxxxxx'
      GO TO 615
670 PC(I) = PC(I) / SBASE
      QC(I) = QC(I) / SBASE
      PG(I) = PG(I) / SBASE
680 INDL = 0

```

Se define el menú para interconectar el nuevo nodo.

```

700 CALL INIMEN
      ENCABE = ' MENU DE INTERCONEXIONES DEL NUEVO NODO '
      LISTOP(1) = ' REGRESAR A MENU DE ADICION DE DATOS '
      LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
      LISTOP(3) = ' ADICION DE LINEAS DE TRANSMISION '
      LISTOP(4) = ' ADICION DE TRANSFORMADORES '
      LISTOP(5) = ' ADICION DE COMPENSACION EN DERIVACION '
      OPCION = 1
      CALL MENU (OPCION)
      NPCION = OPCION
      NODOS = NN
      JX = I
      IF (OPCION.EQ.1.AND.INDL.EQ.1) GO TO 18
      IF (OPCION.EQ.2.AND.INDL.EQ.1) GO TO 15
      IF (OPCION.NE.3) GO TO 702

```

Se interconectará el nuevo nodo mediante líneas de transmisión.

```

CALL INIMEN
VALCHA(3) = NAME1

```

```

VALCHA(4) = NAME2
GO TO 103
702 IF (OPCION.NE.4) GO TO 710

```

Se interconectará el nuevo nodo mediante líneas de transformadores. Se pregunta si el nuevo nodo estará del lado del tap o del lado opuesto.

```

704 CALL CUADRO
ANSWER = 'SI'
WRITE (*, 705) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE, ANSWER, ESC
705 FORMAT(A, '[10;5H', A, A4, '¿ EL NUEVO NODO ESTA AL LADO DEL TAP ',
* '[ SI, NO ] ? ', /, A, '[10;68H', A, A4, A2, A, '[10;68H', $)
READ (*, 592) RESPUE
CALL SINO (RESPUE, ANSWER)
IF (RESPUE.EQ.'SI') ANSWER = 'SI'
IF (RESPUE.EQ.'NO') ANSWER = 'NO'
IF (ANSWER.NE.'SI'.AND.ANSWER.NE.'NO') GO TO 704
CALL INIMEN
JX = I
VALCHA(3) = NAME1
VALCHA(4) = NAME2
IF (ANSWER.EQ.'SI') GO TO 203
JY = I
VALCHA(4) = NAME1
VALCHA(3) = NAME2
GO TO 203
710 IF (OPCION.NE.5) GO TO 700

```

Existirá compensación en derivación en el nuevo nodo.

```

JX = I
CALL INIMEN
VALCHA(3) = NAME1
GO TO 455
C***** SE MODIFICAN LINEAS, TRANSF., REACTORES/CAPACITORES, GEN. Y CARGA

```

Se modifican datos en líneas de transmisión.

```

1100 JY = 0
1101 CALL CUADRO
WRITE (*, 1102) ESC, WHITEFORE, ESC
1102 FORMAT(A, A4, A, '[24;12H',
* '***Oprima la tecla de "entrada" para salir de esta opción...')
WRITE (*, 1103) ESC, ESC, YELFORE
1103 FORMAT (A, '[6;25H', A, A4, 'CAMBIOS EN LINEAS DE TRANSMISION')

```

Se leen y se verifican los nodos de envío y de recepción de la línea.

```

WRITE (*, 1105) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
1105 FORMAT(A, '[10;10H', A, A4, '¿ NOMBRE DEL NODO DE ENVIO ',
* '[máximo 8 caracteres] ? ', /, A, '[10;66H', A, A4, $)
READ (*, 110) NAME1

```

Se regresa a menú principal de modificación en elementos o nodos.

```

      IF (NAME1.EQ.' ') GO TO 30
      IF (JY.NE.0) GO TO 1180
1106 WRITE (*, 1107) ESC
1107 FORMAT (A, '[19;10H', '
      *
1160 WRITE (*, 517) ESC
      WRITE (*, 1165) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
1165 FORMAT(A, '[10;10H', A, A4, '¿ NOMBRE DEL NODO DE RECEPCION ',
      * '[máximo 8 caracteres] ? ',/, A, '[10;66H', A, A4, $)
      READ (*, 110) NAME2

```

Se regresa a menú principal de modificación en elementos o nodos.

```

      IF (NAME2.EQ.' ') GO TO 30
1166 WRITE (*, 1107) ESC

```

Se revisan los nombres de nodos leídos.

```

      ITIP = 1
      CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME2, IEX)

```

Se regresa a leer nuevamente el nombre de nodo, por no existir en la lista de nombres DNOMB(.).

```

      IF (JX.EQ.0) GO TO 1100
      IF (JY.EQ.0) GO TO 1160

```

Se leyeron dos nombres de nodos correctamente. Ahora, se verifica que el elemento de transmisión exista.

```

1180 KC1 = INICPR(JX)
1186 IF (KC1.EQ.0) GO TO 1205
      I = IPOSPR(KC1)
      IF (I.LT.0) GO TO 1188
      IF (NE(I).EQ.JX.AND.NR(I).EQ.JY) GO TO 1220
      IF (NE(I).EQ.JY.AND.NR(I).EQ.JX) GO TO 1220
1188 KC1 = IFINPR(KC1)
      GO TO 1186
1205 CONTINUE
      WRITE (*, 1107) ESC
      WRITE (*, 1210) ESC, NAME1, NAME2
1210 FORMAT(A, '[19;10H', 'EL ELEMENTO CON LOS NODOS ', A8,
      * ' Y ', A8, ' NO EXISTE...')
      WRITE (*, 517) ESC
      GO TO 1100
1220 CONTINUE

```

La línea existe y se prepara el menú para modificar sus datos. A NR(I) se le cambia el signo para evitar acceder el elemento de transmisión posteriormente.

```

      NR(I) = - NR(I)
1235 CALL INIMEN

```

```

DO 1236 K = 4, 6
1236     TIPVAL(K) = 'REA'
        VALNUM(4) = R(I)
        VALNUM(5) = X(I)
        VALNUM(6) = YPQ2E(I)
        ENCADE = '   MODIFICACIONES A PARAMETROS DE LINEAS   '
        ALTERN = '   ( CUAL OPCION DESEA [1'
        LISTOP(1) = '   REGRESAR A MENU DE MODIFICACIONES   '
        LISTOP(2) = '   REGRESAR A MENU PRINCIPAL DE CAMBIOS   '
        LISTOP(3) = '   SELECCIONAR      OTRA      LINEA      '
        LISTOP(4) = '   RESISTENCIA  SERIE  DE  LA  LINEA  '
        LISTOP(5) = '   REACTANCIA   SERIE  DE  LA  LINEA   '
        LISTOP(6) = '   MITAD DE LA ADMITANCIA EN DERIVACION   '
        OPCION = 3
        CALL  MENU  (OPCION)

```

Se actualizan datos del elemento.

```

R(I) = VALNUM(4)
X(I) = VALNUM(5)
YPQ2E(I) = VALNUM(6)
IF (OPCION.NE.3) GO TO 1240
WRITE (*, 1102) ESC, WHITEFORE, ESC
WRITE (*, 1103) ESC, ESC, YELFORE

```

Se regresa a seleccionar otra línea.

```

GO TO 1100

```

Se restablece la información de nodos de recepción de las líneas.

```

1240 DO 1290 J = 1, LINES
1290     NR(J) = ABS(NR(J))

```

Se regresa a menú principal de modificación en elementos o nodos.

```

IF (OPCION.EQ.1) GO TO 30

```

Se regresa a menú principal del subprograma.

```

IF (OPCION.EQ.2) GO TO 15
C***** SE MODIFICAN DATOS DE TRANSFORMADORES

```

Se modifican datos de transformadores. Se leen nombres de los nodos extremos.

```

1300 CALL CUADRO
        JY = 0
1301 WRITE (*, 1102) ESC, WHITEFORE, ESC
        WRITE (*, 1302) ESC, ESC, YELFORE
1302 FORMAT (A, '[6;16H', A, A4,
        *           'MODIFICACIONES EN DATOS DE TRANSFORMADORES')
1303 WRITE (*, 1305) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
1305 FORMAT(A, '[10;10H', A, A4, '¿ NOMBRE DEL NODO AL LADO DEL TAP ',
        *           '[máximo 8 caracteres] ? ', '/', A, '[10;68H', A, A4, '$)
        READ (*, 110) NAME1

```

Se regresa a menú principal de modificación en elementos o nodos.

```

IF (NAME1.EQ.' ') GO TO 30
IF (JY.NE.0) GO TO 1360
1320 WRITE (*, 517) ESC
      WRITE (*, 1325) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
1325 FORMAT(A, '[10;10H', A, A4, '¿ NOMBRE DEL NODO CONTRARIO AL TAP ',
* '[máximo 8 caracteres] ? ', /, A, '[10;69H', A, A4, $)
      READ (*, 110) NAME2

```

Se regresa a menú principal de modificación en elementos o nodos.

```

IF (NAME2.EQ.' ') GO TO 30

```

Se revisan los nombres de nodos leídos.

```

ITIP = 1
CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME2, IEX)

```

Se regresa a leer nuevamente el nombre de nodo, por no existir en la lista de nombres DNOMB(.).

```

IF (JX.EQ.0) GO TO 1300
IF (JX.NE.0) GO TO 1360

```

Se leyeron dos nombres de nodos correctamente. Ahora, se verifica que el transformador exista. En caso afirmativo, se procede a modificar sus datos. En caso que no exista, se imprime el mensaje correspondiente y se regresa a leer nombres de nodos de otro transformador.

```

CALL CUADRO
GO TO 1320

```

IND funciona como bandera para indicar el elemento que se modificará.

```

1360 IND = 0
      DO 1370 J = 1, NTAPS
            JC = NETR(J)
            JS = NRTR(J)
            T = TAP(J)
            IF (JX.NE.JC.OR.JY.NE.JS) GO TO 1368
            IF (T.GT.0.0) GO TO 1380

```

El tap es menor que cero y es indicativo de que este transformador ya había sido seleccionado para modificarle sus parámetros. Aquí, se presenta la oportunidad de seleccionarlo nuevamente.

```

1361      CALL CUADRO
            T2 = ABS(T)
            ANSWER = 'NO'
            WRITE (*, 1362) ESC, YELFORE, ESC, ESC, J, ESC, NAME1, ESC,
*              NAME2, ESC, T2, ESC, ESC, ESC, WHITEFORE, ESC, ANSWER, ESC
1362      FORMAT(A, A4, A, '[6;20H', 'EL TRANSFORMADOR CON LOS DATOS:',
*              /, A, '[8;20H', 'TRANSFORMADOR No. : ', I3,
*              /, A, '[10;20H', 'NODO LADO DEL TAP : ', A8,

```



```

*           /,A,'[12;20H','NODO LADO OPUESTO : ',A8,
*           /,A,'[14;20H','VALOR DEL TAP : ',F8.5,
*           /,A,'[16;20H','YA HA SIDO REVISADO... ',/,A,
*           '[19;20H','¿ DESEA MODIFICARLOS NUEVAMENTE [SI, NO] ? ',
*           A,A4,A,'[19;64H',A2,A,'[19;64H',\$)
READ(*, 592) RESPUE
CALL SINO (RESPUE, ANSWER)
IF (RESPUE.EQ.'NO') GO TO 1365
IF (RESPUE.EQ.'SI') GO TO 1367
GO TO 1361
1365      IND = IND + 1
          GO TO 1370
1367      TAP(J) = ABS(T)
          GO TO 1380
1368      IF (JX.NE.JS.OR.JY.NE.JC) GO TO 1370
          WRITE (*, 1379) ESC, ESC, DNOMB(JY), DNOMB(JX)
          WRITE (*, 1369) ESC
1369      FORMAT (A,'[24;15H',
*           ,
*           ' )
          CALL ENCA
          GO TO 1300
1370      CONTINUE
          IF (IND.NE.0) GO TO 1300

```

El transformador seleccionado no existe.

```

WRITE (*, 1210) ESC, NAME1, NAME2
WRITE (*, 517) ESC
CALL ENCA
GO TO 1300
1376      FORMAT(A,'[19;10H','ESTE TRANSFORMADOR YA HA SIDO REVISADO...')
1379      FORMAT(A,'[19;10H','LOS NODOS DE ESTE TRANSF. ESTAN INVERTIDOS:',
*           */,A,'[20;10H','NODO LADO TAP: ',A8,3X,'NODO CONTRARIO TAP : ',A8)

```

Ya está definido el transformador cuyos parámetros serán modificados.

```

1380      CALL CUADRO
          WRITE (*, 1382) ESC, ESC, J, ESC, NAME1, ESC, NAME2
1382      FORMAT (A,'[10;15H','SE REVISAS EL TRANSFORMADOR CON LOS DATOS:'
*           ,/,A,'[12;20H','TRANSFORMADOR          No. : ',I3
*           ,/,A,'[14;20H','NODO AL LADO DEL TAP : ',A8
*           ,/,A,'[16;20H','NODO CONTRARIO AL TAP : ',A8)
          CALL ENCA
1385      CALL INIMEN
          TIPVAL(3) = 'CHA'
          TIPVAL(4) = 'REA'
          TIPVAL(5) = 'REA'

```

Se busca el valor de la reactancia serie del transformador.

```

DO 1386  K = 1, NLINES
          IF (JX.NE.NE(K).OR.JY.NE.NR(K)) GO TO 1386
          IF (IND.EQ.0) VALNUM(4) = X(K)
          IF (IND.EQ.0) GO TO 1387

```

```

                IND = IND - 1
1386 CONTINUE
1387 VALNUM(5) = TAP(J)
    ENCABE = ' MODIFICACION A DATOS DE TRANSFORMADORES '
    ALTERN = ' ( CUAL OPCION DESEA [1'
    LISTOP(1) = ' REGRESAR A MENU DE MODIFICACIONES '
    LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
    LISTOP(3) = ' CIRCUITO II EQUIVALENTE DEL ELEMENTO '
    LISTOP(4) = ' REACTANCIA SERIE DEL TRANSFORMADOR '
    LISTOP(5) = ' VALOR DEL TAP DEL TRANSFORMADOR '
    OPCION = 1
    CALL MENU (OPCION)
    X(K) = VALNUM(4)
    TAP(J) = VALNUM(5)
    TAP(J) = - TAP(J)
    IND = 1388
    GO TO 310
1388 IF (OPCION.EQ.1) GO TO 1300
1389 IF (OPCION.EQ.2) GO TO 15
    IF (OPCION.EQ.3) GO TO 1400
C***** SE LEE EL NUEVO CIRCUITO II DEL TRANSFORMADOR

```

Se ha optado por modificar directamente los parámetros del circuito II del transformador.

```

1400 CALL INIMEN
    DO 1402 L = 3, 8
1402     TIPVAL(L) = 'REA'
    VALNUM(3) = YPQ2E(K)
    VALNUM(4) = GPQ2E(K)
    VALNUM(5) = YPQ2R(K)
    VALNUM(6) = GPQ2R(K)
    VALNUM(7) = X1(K)
    VALNUM(8) = R1(K)
    ENCABE = ' CAMBIOS A CIRCUITOS II DE TRANSFORMADORES '
    LISTOP(1) = ' REGRESAR A MENU DE MODIFICACIONES '
    I = K
    GO TO 370
1410 IF (TAP(J).GT.0.0) TAP(J) = - TAP(J)
    IF (OPCION.EQ.2) GO TO 15
    GO TO 30
C***** SE MODIFICAN DATOS DE COMPENSADORES EN DERIVACION.
1450 IF (NSHNTS.NE.0) GO TO 1455

```

No existe compensación en derivación en el sistema eléctrico, por lo cual esta opción no es válida. Se imprime el mensaje y se regresa al menú de modificación de datos.

```

CALL CUADRO
WRITE (*, 410) ESC, ESC
CALL ENCA
GO TO 35

```

Existe compensación en derivación y se lee el nombre del nodo donde se desea modificar su compensación.

```

1455 CALL CUADRO
      WRITE (*, 1102) ESC, WHITEFORE, ESC
      WRITE (*, 1457) ESC, ESC, YELFORE
1457 FORMAT (A, '[6;15H', A, A4,
*       'CAMBIOS A PARAMETROS DE COMPENSADORES EN DERIVACION')
1459 WRITE (*, 1460) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
1460 FORMAT(A, '[10;5H', A, A4, '¿ NOMBRE DEL NODO DEL COMPENSADOR ',
*       '[máximo 8 caracteres] ? ', /, A, '[10;68H', A, A4, $)
      READ (*, 110) NAME1

```

Se regresa a menú de modificación de datos en nodos o elementos.

```

      IF (NAME1.EQ.' ') GO TO 35
      WRITE (*, 1107) ESC

```

Se verifica el nombre del nodo leído.

```

      ITIP = 2
      CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME1, IEX)
      IF (JX.EQ.0) GO TO 1455

```

El nombre de nodo leído es correcto. Ahora, se verifica que contenga compensación en derivación.

```

      DO 1472 I = 1, NSHNTS
1472      IF (NSHUNT(I).EQ.JX) GO TO 1480
      WRITE (*, 1475) ESC, NAME1
1475 FORMAT(A, '[19;10H', 'EL NODO ', A8,
*       ' NO TIENE COMPENSACION EN DERIVACION')
      WRITE (*, 507) ESC
      GO TO 1459

```

Existe compensación en derivación en el nodo seleccionado, por lo que se procede a modificar el valor de la susceptancia del compensador.

```

1480 WRITE (*, 1483) ESC, YELFORE, ESC, ESC, NAME1, ESC, WHITEFORE,
*       SHUNT(I), ESC, ESC, YELFORE, ESC, WHITEFORE
1483 FORMAT(A, A4, A, '[13;6H', 'SE MODIFICA EL VALOR DEL COMPENSADOR',
*       ' EN EL NODO', /, A, '[13;55H', A8, ' : ', A, A4, F9.5, /, A,
*       '[17;20H', A, A4, '¿ NUEVO VALOR DEL COMPENSADOR ? ', A, A4, $)
      READ (*, 1485, ERR=1480) SH
      IF (SH.NE.0.0) SHUNT(I) = SH
1485 FORMAT (F14.9)
      GO TO 1455

```

Se modifican datos de carga nodal.

```

1500 CALL INIMEN
      ENCABE = ' MODIFICACIONES A DATOS DE CARGA : '
      LISTOP(1) = ' REGRESAR A MENU DE MODIFICACIONES '
      LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
      LISTOP(3) = ' POTENCIA REAL POR NODO '
      LISTOP(4) = ' POTENCIA REACTIVA POR NODO '

```

```

LISTOP(5) = ' A FACTOR DE POT. CONSTANTE POR NODO '
LISTOP(6) = ' POTENCIA REAL POR ISLA '
LISTOP(7) = ' POTENCIA REACTIVA POR ISLA '
LISTOP(8) = ' A FACTOR DE POT. CONSTANTE POR ISLA '
1501 OPCION = 1

```

Se selecciona una de las opciones listadas.

```
CALL MENU (OPCION)
```

Se regresa a menú de modificaciones de datos ya existentes.

```
IF (OPCION.EQ.1) GO TO 30
```

Se regresa a menú principal del subprograma.

```

IF (OPCION.EQ.2) GO TO 15
IF (OPCION.NE.3) GO TO 1502
IND = 0
STRING='MODIFICACION DE POTENCIA REAL DE CARGA POR NODO'
GO TO 1505
1502 IF (OPCION.NE.4) GO TO 1503
IND = 1
STRING='MODIFICACION DE POT. REACTIVA DE CARGA POR NODO'
GO TO 1505
1503 IF (OPCION.EQ.5) GO TO 1580
IF (OPCION.EQ.6) GO TO 1620
IF (OPCION.EQ.7) GO TO 1650
IF (OPCION.EQ.8) GO TO 1680
GO TO 1501
1505 CALL CUADRO

```

Se modifica potencia real o reactiva de carga por nodo. Aquí, se lee el nombre de éste.

```

WRITE (*, 1102) ESC, WHITEFORE, ESC
WRITE (*, 1507) ESC, ESC, YELFORE, STRING
1507 FORMAT (A, ' [6;15H', A, A4, A60)
1509 WRITE (*, 510) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
READ (*, 110) NAME1

```

Si NAME1 = ' ' se regresa a menú de modificación de datos de carga.

```

IF (NAME1.EQ.' ') GO TO 1501
WRITE (*, 1107) ESC

```

Se verifica el nombre del nodo leído. Si no es correcto (JX = 0), se regresa a leer nuevamente otro nombre de nodo.

```

ITIP = 2
CALL VERIF (NODOS, ITIP, JX, JX, KONT, DNOMB, NAME1, NAME1, IEX)
IF (JX.EQ.0) GO TO 1505

```

El nombre de nodo leído es correcto, por lo que se lee ahora el nuevo valor de potencia real o reactiva, según el caso.

```

1520 SH = PC(JX) * SBASE
      IF (IND.EQ.1) SH = QC(JX) * SBASE
      IF (IND.EQ.0) WRITE(*,1523) ESC, YELFORE, ESC, ESC, NAME1, ESC,
*          WHITEFORE, SH, ESC, ESC, YELFORE, ESC, WHITEFORE
1523 FORMAT (A,A4,A,'[13;6H','SE MODIFICA LA POTENCIA REAL DE CARGA ',
* 'EN EL NODO',/,A,'[13;55H',A8,' : ',A,A4,F9.2,/,A,'[17;20H',
* A,A4,'¿ NUEVA POTENCIA REAL DE CARGA (MW) ? ',A,A4,$)
      IF (IND.EQ.1) WRITE(*,1524) ESC, YELFORE, ESC, ESC, NAME1, ESC,
*          WHITEFORE, SH, ESC, ESC, YELFORE, ESC, WHITEFORE
1524 FORMAT (A,A4,A,'[13;6H','SE MODIFICA LA POT. REACTIVA DE CARGA ',
* 'EN EL NODO',/,A,'[13;55H',A8,' : ',A,A4,F9.2,/,A,'[17;20H',
* A,A4,'¿ NUEVA POT. REACTIVA DE CARGA (MVAR) ? ',A,A4,$)
      READ (*,1485,ERR=1520) SH

```

Si SH = 0, no se asigna este valor a la variable correspondiente.
 Si SH ≠ 0, se asignará a la variable correspondiente.

```

      IF (SH.NE.0.0.AND.IND.EQ.0) PC(JX) = SH / SBASE
      IF (SH.NE.0.0.AND.IND.EQ.1) QC(JX) = SH / SBASE
      GO TO 1505

```

Se modificará la carga real y reactiva por un mismo factor. Aquí, se lee el nombre del nodo.

```

1580 STRING = 'SE MODIFICA CARGA A FACTOR DE POT. CONSTANTE POR NODO'
1584 CALL CUADRO
      WRITE (*, 1102) ESC, WHITEFORE, ESC
      WRITE (*, 1507) ESC, ESC, YELFORE, STRING
1589 WRITE (*, 510) ESC, ESC, YELFORE, ESC, ESC, WHITEFORE
      READ (*, 110) NAME1

```

Si NAME1 = ' ' se regresa a menú de modificación de datos de carga.

```

      IF (NAME1.EQ.' ') GO TO 1501
      WRITE (*, 1107) ESC

```

Se verifica el nombre del nodo leído. Si no es correcto (JX = 0), se regresa a leer nuevamente otro nombre de nodo.

```

      ITIP = 2
      CALL VERIF (NODOS, ITIP, JX, JX, KONT, DNOMB, NAME1, NAME1, IEX)
      IF (JX.EQ.0) GO TO 1584

```

El nombre de nodo es correcto. Entonces, se inicializa el factor en 1.0 y se procede a leer su nuevo valor.

```

1600 SH = 1.0
      WRITE (*,1603) ESC, YELFORE, ESC, ESC, NAME1, ESC, WHITEFORE,
*          PC(JX)*SBASE, QC(JX)*SBASE, ESC, ESC, YELFORE, ESC, WHITEFORE
1603 FORMAT(A,A4,A,'[13;5H','SE MODIFICA LA CARGA EN EL NODO',/,A,
*'[13;38H',A8,' : ',A,A4,F8.2,' MW',/,A,A4,F8.2,' MVAR',/,A,'[17;15H',
*A,A4,'¿ FACTOR DE VARIACION A LA CARGA (NUMERO REAL) ? ',A,A4,$)
      READ (*,1485,ERR=1600) SH

```

Si SH = 0, no se modifica la carga en el nodo.
 Si SH ≠ 0, se modifica la carga en el nodo.

```

    IF (SH.NE.0.0) PC(JX) = SH * PC(JX)
    IF (SH.NE.0.0) QC(JX) = SH * QC(JX)
    GO TO 1584
  1620 STRING='SE MODIFICA POTENCIA REAL DE CARGA POR ISLA'
```

Se modifica la potencia real de carga en todos los nodos de una isla.
 El factor de modificación se guarda en FPCISLA(KI), siendo KI el número de isla seleccionado.

```

    CALL CUADRO
    WRITE (*, 1507) ESC, ESC, YELFORE, STRING
    KI = 1
```

Si el total de islas en el sistema es 1, se pasa directamente a leer el factor para modificar la potencia real de carga del sistema.

```

    IF (NISLAS.EQ.1) GO TO 1625
```

El total de islas es mayor que 1, por lo que se lee el número de isla en donde se modifica la potencia real de carga.

```

    WRITE (*, 1623) ESC, ESC, YELFORE, NISLAS, ESC, ESC, WHITEFORE
  1622 FORMAT(A, '[10;8H', A, A4,
    * '¿ FACTOR (Numero Real) PARA MODIFICAR POT. REAL EN LA ISLA ?'
    *      ,/, A, '[10;69H', A, A4, F5.3, A, '[10;69H', $)
  1623 FORMAT(A, '[10;15H', A, A4, '¿CUAL ISLA DESEA MODIFICAR ',
    *      '[ 1, ..., ', I2, ' ] ?      ',/, A, '[10;64H', A, A4, $)
    READ (*, 1624, ERR=1620) KI
  1624 FORMAT(I2)
```

Se verifica que el número de isla leído esté correcto.

```

    IF (KI.EQ.0.OR.KI.GT.NISLAS) GO TO 1620
```

El número de isla es correcto, por lo que se procede a leer el factor de modificación de la potencia real de carga en esta isla.

```

  1625 SH = FPCISLA(KI)
    WRITE (*, 1622)ESC, ESC, YELFORE, ESC, ESC, WHITEFORE, SH, ESC
    READ (*, 1485, ERR=1625) SH
```

Si SH ≠ 0, se guarda el factor en FPCISLA(KI).

```

    IF (SH.NE.0.0) FPCISLA(KI) = SH
    GO TO 1501
  1650 STRING='SE MODIFICA POTENCIA REACTIVA DE CARGA POR ISLA'
```

Esta secuencia es idéntica a la de modificación de potencia real de carga por isla. Aquí, se lee un factor para modificar la potencia reactiva de carga por isla. Este valor se guarda en FQCISLA(KI).

```

    CALL CUADRO
    WRITE (*, 1507) ESC, ESC, YELFORE, STRING
```

```

KI = 1
IF (NISLAS.EQ.1) GO TO 1665
1655 WRITE (*,1623)ESC,ESC,YELFORE,NISLAS,ESC,ESC,WHITEFORE
1662 FORMAT(A,'[10;10H',A,A4,'¿ FACTOR DE MODIFICACION DE POT. ',
* 'REACTIVA EN LA ISLA ? ',/,A,'[10;67H',A,A4,F5.3,A,'[10;67H',,$)
READ (*,1624,ERR=1655) KI
IF (KI.EQ.0.OR.KI.GT.NISLAS) GO TO 1650
1665 SH = FQCISLA(KI)
WRITE (*,1662)ESC,ESC,YELFORE,ESC,ESC,WHITEFORE,SH,ESC
READ (*,1485,ERR=1665) SH
IF (SH.NE.0.0) FQCISLA(KI) = SH
GO TO 1501
1680 STRING='SE MODIFICA POT. DE LA ISLA A F.P. CONSTANTE'

```

Con la misma secuencia de los dos casos anteriores, se lee un factor para modificar por igual las potencias real y reactiva de una isla.

```

CALL CUADRO
WRITE (*,1507) ESC,ESC,YELFORE,STRING
KI = 1
IF (NISLAS.EQ.1) GO TO 1685
1681 WRITE (*,1623) ESC,ESC,YELFORE,NISLAS,ESC,ESC,WHITEFORE
1682 FORMAT(A,'[10;10H',A,A4,'¿ FACTOR DE MODIFICACION DE POTENC',
* 'IA EN LA ISLA ? ',/,A,'[10;67H',A,A4,F5.3,A,'[10;67H',,$)
READ (*,1624,ERR=1681) KI
IF (KI.EQ.0.OR.KI.GT.NISLAS) GO TO 1680
1685 SH = FPCISLA(KI)
WRITE (*,1682) ESC,ESC,YELFORE,ESC,ESC,WHITEFORE,SH,ESC
READ (*,1485,ERR=1685) SH
IF (SH.EQ.0.0) SH = 1.0
FQCISLA(KI) = SH
FPCISLA(KI) = SH
GO TO 1501
C***** SE MODIFICAN DATOS EN NODOS DE VOLTAJE CONTROLADO.

```

Modificación de parámetros en nodos de voltaje controlado. Se lee el nombre del nodo.

```

1700 CALL CUADRO
WRITE (*,1102) ESC,WHITEFORE,ESC
WRITE (*,1707) ESC,ESC,YELFORE
1707 FORMAT (A,'[6;12H',A,A4,
* 'MODIFICACION DE PARAMETROS EN NODOS DE VOLTAJE CONTROLADO')
1709 WRITE (*,1710) ESC,ESC,YELFORE,ESC,ESC,WHITEFORE
1710 FORMAT(A,'[10;5H',A,A4,'¿ NOMBRE DEL NODO DE GENERACION ',
* '[máximo 8 caracteres] ? ',/,A,'[10;68H',A,A4,$)
READ (*,110) NAME1

```

Si NAME1 = ' ' se regresa a menú de modificación de datos.

```
IF (NAME1.EQ.' ') GO TO 30
```

Se verifica el nombre del nodo leído.

```
ITIP = 2
```

```
CALL VERIF (NODOS, ITIP, JX, JX, KONT, DNOMB, NAME1, NAME1, IEX)
IF (JX.EQ.0) GO TO 1700
```

El nombre de nodo es correcto. Ahora, se verifica que sea un nodo P-V.

```
DO 1711 L = 1, NNGEN
1711 IF (NPV(L).EQ.JX) GO TO 1715
```

El nodo no es generador. Se imprime mensaje y regresa a leer otro nodo.

```
WRITE (*,1712) ESC, NAME1
1712 FORMAT (A,'[19;10H','EL NODO ',A8,' NO ES DE GENERACION... ')
WRITE (*, 507) ESC
GO TO 1709
```

Se tiene un nodo generador. Se prepara menú para modificar sus datos.

```
1715 CALL INIMEN
      VALNUM(4) = PG(JX) * SBASE
      VALNUM(5) = QGMAX(L) * SBASE
      VALNUM(6) = QGMIN(L) * SBASE
      VALNUM(7) = VESP(L)
      DO 1716 I = 4, 7
1716 TIPVAL(I) = 'REA'
      TIPVAL(8) = 'CHA'
      VALCHA(3) = DNOMB(ABS(NAU(L)))
1717 VALCHA(8) = 'M-S'
      IF (TIPGEN(L).EQ.1) VALCHA(8) = 'CEV'
      ENCABE = ' MODIFICACION A DATOS DE GENERADORES : '
      LISTOP(1) = ' REGRESAR A MENU DE MODIFICACIONES '
      LISTOP(2) = ' REGRESAR A MENU PRINCIPAL DE CAMBIOS '
      LISTOP(3) = ' NODO ALTA TENSION CONECTADO A GENERADOR '
      LISTOP(4) = ' POTENCIA REAL DE GENERACION '
      LISTOP(5) = ' LIMITE MAXIMO POT. REACTIVA GENERADA '
      LISTOP(6) = ' LIMITE MINIMO POT. REACTIVA GENERADA '
      LISTOP(7) = ' VOLTAJE DE GENERACION '
      LISTOP(8) = ' CAMBIO DE TIPO DE GENERADOR [CEV, M-S] '
1720 OPCION = 1
      CALL MENU (OPCION)
      NAME2 = VALCHA(3)
```

Se verifica el nombre del nodo de alta tensión asociado al nodo leído.

```
ITIP = 1
CALL VERIF (NODOS, ITIP, JX, JY, KONT, DNOMB, NAME1, NAME2, IEX)
```

Si JY = 0, el nombre es incorrecto. Entonces, se inicializa VALCHA(3) y se regresa al menú de modificación de datos de generadores.

```
IF (JY.EQ.0) VALCHA(3) = '.....'
IF (JY.EQ.0) GO TO 1720
```

Se actualizan variables del nodo de generación leído.

```
PG(JX) = VALNUM(4) / SBASE
QGMAX(L) = VALNUM(5) / SBASE
```



```

QGMIN(L) = VALNUM(6) / SBASE
IF (VALNUM(7).GE.0.75) VESP(L) = VALNUM(7)
IF (VALCHA(8).EQ.'CEV'.OR.VALCHA(8).EQ.'M-S') GO TO 1740
GO TO 1717
1740 TIPGEN(L) = 0
NAU(L) = JY
IF (VALCHA(8).NE.'CEV') GO TO 1750
TIPGEN(L) = 1
NAU(L) = - JY
IF (NAME1.NE.NAME2) GO TO 1750

```

Cuando el nodo P-V es un compensador estático de VARs, el nodo de alta tensión no puede llamarse igual al primero. En caso de ser así, se reinicializa VALCHA(3) y se regresa al menú de modificación de datos de generadores.

```

VALCHA(3) = '.....'
GO TO 1720

```

Se regresa a menú principal de cambios en datos de elementos o nodos.

```

1750 IF (OPCION.EQ.1) GO TO 30

```

Se regresa a menú principal del subprograma.

```

IF (OPCION.EQ.2) GO TO 15
C ***** SE GRABA ARCHIVO PARA OTRO CASO DE ESTUDIO DE FLUJOS

```

Se inicializa ARCH1 con el nombre de archivo de 'default'.

```

2000 ARCH1 = 'NUEVOSDA.DAT'

```

Se lee el nombre de archivo.

```

CALL CUADRO
WRITE (*,2010) ESC, ESC, WHITEFORE, ESC, ARCH1, ESC
2010 FORMAT (A, '[6;3H', 'NOMBRE DEL NUEVO ARCHIVO DE DATOS ',
* '[maximo 40 caracteres] ? ', A, A4, /, A, '[8;5H', A40, A, '[6;5H', $)
READ (*, 2015) ARCH3
2015 FORMAT (A40)

```

Si ARCH3 = ' ', el nombre adoptado es el 'default'.

```

IF (ARCH3.EQ.' ') ARCH3 = ARCH1

```

Se abre el archivo para empezar a grabar.

```

OPEN (UNIT = 8, FILE = ARCH3 )

```

Se presenta la opción de modificar la descripción del archivo de datos actual (primeros dos renglones del mismo).

```

STRING = ' PRIMER '
LETR3 = ' '
WRITE (*,2020)ESC, ESC, YELFORE, STRING, ESC, LETR1, ESC, WHITEFORE, ESC
2020 FORMAT(A, '[10;10H', A, A4, 'ESCRIBA', A9, 'RENGLON DE DESCRIPCION : ',
* /, A, '[12;5H', A70, /, A, A4, A, '[12;5H', $)

```

```
READ (*, 2060) LETR3
```

Si LETR3 = ' ', se graba el renglón que se tenía anteriormente. De otro modo se graba lo que se leyó.

```
IF (LETR3.EQ.' ') WRITE (8,2060) LETR1
IF (LETR3.NE.' ') WRITE (8,2060) LETR3
```

Se hace lo mismo con el segundo renglón de descripción de archivo.

```
STRING = ' SEGUNDO '
LETR3 = ' '
WRITE (*,2020)ESC,ESC,YELFORE,STRING,ESC,LETR2,ESC,WHITEFORE,ESC
READ (*, 2060) LETR3
IF (LETR3.EQ.' ') WRITE (8, 2060) LETR2
IF (LETR3.NE.' ') WRITE (8, 2060) LETR3
2060 FORMAT (A70)
```

Se imprimen los nombres de nodos y potencias de carga. Se verifica que cada nodo esté dado de alta en el sistema eléctrico.

```
DO 2100 I = 1, NODOS
      DO 2070 K = 1, KFUERA
2070      IF (NODFUE(K).EQ.I) GO TO 2100
          WRITE (8,2080)DNOMB(I),PC(I)*SBASE,QC(I)*SBASE
2080      FORMAT (2X,A8,2(2X,F8.2))
2100 CONTINUE
```

Se imprime el cero para indicar final de bloque de datos.

```
WRITE (8,2195)
```

Se imprimen datos de nodos de generación.

```
DO 2110 I = 1, NNGEN
      J = ABS(NAU(I))
2110      IF (NPV(I).GT.0) WRITE(8,2120)DNOMB(NPV(I)),DNOMB(J),
          *PG(NPV(I))*SBASE,QGMAX(I)*SBASE,QGMIN(I)*SBASE,VESP(I),TIPGEN(I)
2120      FORMAT (2X,A8,2X,A8,2X,3(F8.2,2X),F8.6,1X,I1)
```

Se imprime el cero para indicar final de bloque de datos.

```
WRITE (8,2195)
```

Se imprimen datos de transformadores.

```
DO 2140 I = 1, NTAPS
      IF (NETR(I).LT.0) GO TO 2140
      DO 2135 J = 1, N LINES
          IF (NETR(I).NE.NE(J).OR.NRTR(I).NE.NR(J)) GO TO 2135
              WRITE(8,2130)DNOMB(NE(J)),DNOMB(NR(J)),R(J),X(J),
          *
              YPQ2E(J)*0.0, ABS(TAP(I))
2130      FORMAT (2X,A8,2X,A8,4(2X,F7.5))
          NR(J) = - NR(J)
          GO TO 2140
2135 CONTINUE
```

Se imprimen datos de líneas de transmisión.

```
2140 CONTINUE
      T = 0.0
      DO 2160 I = 1, NLINES
2160   IF (NE(I).GT.0.AND.NR(I).GT.0) WRITE (8,2130) DNOMB(NE(I)),
      *                               DNOMB(NR(I)),R(I),X(I),YPQ2E(I),T
```

Se imprime el cero para indicar final de bloque de datos.

```
      WRITE (8,2195)
```

Se imprimen datos de compensación en derivación.

```
      DO 2170 I = 1, NSHNTS
2170   IF (NSHUNT(I).GT.0) WRITE(8,2180)DNOMB(NSHUNT(I)),SHUNT(I)
2180           FORMAT (2X,A8,2X,F8.5)
```

Se imprime el cero para indicar final de bloque de datos.

```
      WRITE (8,2195)
2195 FORMAT (2X,'0')
      CLOSE (UNIT = 8)
```

Se restablece el valor positivo de nodos de recepción en elementos de transmisión.

```
      DO 2197 I = 1, NLINES
2197   NR(I) = ABS(NR(I))
```

Se regresa al menú principal de este subprograma.

```
      GO TO 15
2200 ABORTA = 1
```

Se restablece el valor positivo del tap en transformadores.

```
2220 DO 2230 I = 1, NTAPS
2230   TAP(I) = ABS(TAP(I))
```

Fin del subprograma TOPCAM.

```
      RETURN
      END
```

6. Subprogramas de Enlace

Estos subprogramas tienen la función de ejecutar algunas actividades comunes en subprogramas de entrada, salida o algorítmicos. A continuación, se describe cada uno de los mismos.

a) Subprograma CLEAR

La función de este subprograma es el borrado de pantalla mediante secuencias de escape generadas con la ayuda del Código ASCII.

Este subprograma es accesado mediante la instrucción:

CALL CLEAR

Una descripción de las variables involucradas en este subprograma se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ESC	-	CHARACTER	Substituye a la tecla de 'escape' asignándole el valor equivalente a esta tecla en Código ASCII.
HOME	-	CHARACTER	Se le asigna un valor que al imprimirse en conjunto con ESC se ejecuta un borrado de pantalla y se posiciona el cursor en el primer renglón y primera columna de l pantalla.

El código de este subprograma es el siguiente:

Inicia el subprograma CLEAR.

SUBROUTINE CLEAR

Declaración de variables.

CHARACTER*3 HOME
CHARACTER*1 ESC

PARAMETER (HOME='[2J'])

Se asigna a ESC el valor en código ASCII de la tecla 'escape'.

ESC = CHAR(27)

Se escribe la secuencia de borrado de pantalla y posición del cursor en el primer renglón y la primera columna.

WRITE(6,10)ESC,HOME
10 FORMAT(A1,A3)

Se termina el subprograma CLEAR.

RETURN
END

b) Subprograma ENCA

Este subprograma tiene como función el establecer una pausa en el proceso de simulación y pedir al usuario la presión de la tecla de "entrada" para continuar con el mismo.

La forma de acceder este subprograma es:

CALL ENCA

donde se puede observar que no contiene argumentos. La descripción de sus variables y el código se muestran a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ESC	-	CHARACTER	Substituye a la tecla de 'escape' asignándole el valor equivalente a esta tecla en Código ASCII.
NOM	-	CHARACTER	Se le asigna un valor por medio de una lectura. La pausa en el proceso se genera debido a que se está esperando a que se ejecute la lectura de NOM.

Inicio del subprograma ENCA.

```
SUBROUTINE ENCA
```

C

Se declaran variables.

```
CHARACTER*1 NOM, ESC
```

C

```
ESC = CHAR(27)
```

Se imprime el letrero para teclear "entrada" y se lee la variable NOM.

```
WRITE(6,100) ESC
100 FORMAT(A,'[25;15H',
*           ' ** Oprima la tecla de "entrada" para continuar...'$)
READ(6,102)NOM
102 FORMAT(A1)
```

Una vez leída NOM, se termina este subprograma.

```
RETURN
END
```

c) Subprograma MENU

En este subprograma se selecciona cualquiera de las opciones marcadas en un menú. Además, dependiendo del tipo de opción, es posible leer variables asociadas con tal opción.

La instrucción para entrar a este subprograma es:

```
CALL MENU (OPCION)
```

donde existe sólo un argumento, pero todas las demás variables de entrada y/o salida intervienen en el subprograma a través del bloque común MENU12. La descripción de tales variables se muestra a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
OPCION	E/S	ENTERA	Define la opción a ejecutar en el programa que accesó al subprograma MENU. OPCION siempre entra a MENU con un valor de "default".
VALNUM(15)	E/S	REAL	Se almacenan valores reales asociados con una de las opciones del menú.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
DEPEND(15)	E/S	ENTERA	Define la dependencia entre dos opciones del menú, donde una de ellas aparecerá de acuerdo al valor que tome la otra.
LISTOP(15)	E	CHARACTER	Almacena el texto de cada una de las opciones mostradas en un menú.
ENCABE	E	CHARACTER	Contiene el encabezado que describe al menú actual.
ALTERN	E	CHARACTER	Contiene la pregunta acerca de la opción que se desea ejecutar.
TIPVAL(15)	E	CHARACTER	Contiene una cadena de tres caracteres para cada opción del menú, indicando el tipo de variable asociada.
VALCHA(15)	E/S	CHARACTER	Contiene una cadena de caracteres asociados a cada opción.
ANSWER	-	CHARACTER	Variable auxiliar para leer cadenas de caracteres.
RESPUE	-	CHARACTER	Variable auxiliar para leer cadenas de caracteres.
VAR	-	REAL	Variable auxiliar para leer números reales.

El código del subprograma se muestra a continuación.

Se inicia el subprograma MENU.

SUBROUTINE MENU (OPCION)

Declaración del bloque común MENU12.

COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA

Declaración de variables.

CHARACTER*50	LISTOP(15), ENCABE, ALTERN
CHARACTER*3	TIPVAL(15)
CHARACTER*12	VALCHA(15), ANSWER, RESPUE
CHARACTER	ESC*1, WHITEFORE*4, YELFORE*4
PARAMETER	(YELFORE='[33m', WHITEFORE = '[37m')
INTEGER	OPCION, DEPEND(15)
REAL*8	VALNUM(15), VAR

Se limpia pantalla y se pinta un marco azul.

```
CALL CLEAR
CALL CUADRO
```

C

```
ESC = CHAR(27)
```

C

Se define el número de opciones a intervenir en el menú.

```
N = 0
10 IF (LISTOP(N+1).EQ.' ') GO TO 20
N = N + 1
GO TO 10
```

Se define el espacio para mostrar cada opción en pantalla, de tal forma que queden distribuidas en la misma.

```
20 IF (N.LE.4) KRENGL = 3
IF (N.GT.4.AND.N.LE.8) KRENGL = 2
IF (N.GT.8) KRENGL = 1
IF (N.LT.4) KPRIM = 8
IF (N.GE.4.AND.N.LT.8) KPRIM = 5
IF (N.GE.8) KPRIM = 3
```

Se imprime el encabezado del menú.

```
WRITE (*, 25) ESC, KPRIM, ESC, YELFORE, ENCABE
25 FORMAT(A, '[', I1, '; 12H', A, A4, A50)
```

K es una variable auxiliar para determinar el renglón donde se muestra cada opción en pantalla. IND determina si alguna opción tiene variables asociadas. En este caso, IND = 0, asignado en el siguiente ciclo.

```
K = 1
IND = 1
DO 28 J = 1, N
IF (VALCHA(J).EQ.' '.AND.VALNUM(J).EQ.0.444444) GO TO 28
IND = 0
28 CONTINUE
```

Se determina cuantas opciones son dependientes.

```
M = N
DO 29 J = 1, N
IF (DEPEND(J).LT.0) M = M - 1
29 CONTINUE
```



```

IF(L. LT. 10. AND. VAR. GE. 10. 0. AND. VAR. LT. 100. 0)
*       WRITE(*,55) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, -VAR
IF(L. GE. 10. AND. VAR. GE. 10. 0. AND. VAR. LT. 100. 0)
*       WRITE(*,56) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, -VAR
IF(L. LT. 10. AND. VAR. GE. 0. 0. AND. VAR. LT. 10. 0)
*       WRITE(*,57) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, -VAR
IF(L. GE. 10. AND. VAR. GE. 0. 0. AND. VAR. LT. 10. 0)
*       WRITE(*,58) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, -VAR
VAR = - VAR
GO TO 80
36 IF(L. LT. 10. AND. VAR. LT. 1. 0)
*       WRITE(*,59) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. GE. 10. AND. VAR. LT. 1. 0)
*       WRITE(*,60) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. LT. 10. AND. VAR. GE. 1. 0. AND. VAR. LT. 10. 0)
*       WRITE(*,61) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. GE. 10. AND. VAR. GE. 1. 0. AND. VAR. LT. 10. 0)
*       WRITE(*,62) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. LT. 10. AND. VAR. GE. 10. 0. AND. VAR. LT. 100. 0)
*       WRITE(*,63) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. GE. 10. AND. VAR. GE. 10. 0. AND. VAR. LT. 100. 0)
*       WRITE(*,64) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. LT. 10. AND. VAR. GE. 100. 0. AND. VAR. LT. 1000. 0)
*       WRITE(*,65) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. GE. 10. AND. VAR. GE. 100. 0. AND. VAR. LT. 1000. 0)
*       WRITE(*,66) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. LT. 10. AND. VAR. GE. 1000. 0. AND. VAR. LT. 10000. 0)
*       WRITE(*,67) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
IF(L. GE. 10. AND. VAR. GE. 1000. 0. AND. VAR. LT. 10000. 0)
*       WRITE(*,68) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, VAR
GO TO 80

```

Se muestra una opción con una variable entera asociada.

```

37 IF (TIPVAL(I).NE. 'INT') GO TO 80
IAUX = INT(VALNUM(I))
IF(L. LT. 10. AND. IAUX. LT. 10)
*       WRITE(*,70) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX

```

```

IF(L. GE. 10. AND. IAUX. LT. 10)
*       WRITE(*, 71) ESC, L, ESC, WHITEFORE, I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX
IF(L. LT. 10. AND. IAUX. GE. 10. AND. IAUX. LT. 100)
*       WRITE(*, 72) ESC, L, ESC, WHITEFORE, I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX
IF(L. GE. 10. AND. IAUX. GE. 10. AND. IAUX. LT. 100)
*       WRITE(*, 73) ESC, L, ESC, WHITEFORE, I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX
IF(L. LT. 10. AND. IAUX. GE. 100. AND. IAUX. LT. 1000)
*       WRITE(*, 74) ESC, L, ESC, WHITEFORE, I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX
IF(L. GE. 10. AND. IAUX. GE. 100. AND. IAUX. LT. 1000)
*       WRITE(*, 75) ESC, L, ESC, WHITEFORE, I, ESC, YELFORE,
*       LISTOP(I), ESC, WHITEFORE, IAUX

```

Formatos de impresión.

```

40  FORMAT(A, '[', I1, ';9H', A, A4, A45, ':', A, A4, I2)
41  FORMAT(A, '[', I2, ';9H', A, A4, A45, ':', A, A4, I2)
42  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44)
43  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44)
49  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, A8)
50  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, A8)
51  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.2)
52  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.2)
53  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.3)
54  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.3)
55  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.4)
56  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.4)
57  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
58  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
59  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
60  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
61  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
62  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.5)
63  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.4)
64  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.4)
65  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.3)
66  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.3)
67  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.2)
68  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, F8.2)
70  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I1)
71  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I1)
72  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I2)
73  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I2)
74  FORMAT(A, '[', I1, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I3)
75  FORMAT(A, '[', I2, ';9H', A, A4, I2, A, A4, A44, ':', A, A4, I3)
80  CONTINUE

```

Se hace la pregunta para seleccionar alguna opción.

```

85 I = 10
   RESPUE = '1'
   IF (N.EQ.2) WRITE (*,87)ESC, ESC, YELFORE, ALTERN,
*           M, ESC, WHITEFORE, OPCION, ESC, ESC
87 FORMAT (A, '[23;9H', 9X, A, A4, A30, ', ', I2, ']' ? ', A, A4, I2, /,
*         A, '[23;65H', ', ', A, '[23;59H', '$)
   IF (N.EQ.3) WRITE (*,88)ESC, ESC, YELFORE, ALTERN,
*           M, ESC, WHITEFORE, OPCION, ESC, ESC
88 FORMAT (A, '[23;9H', 9X, A, A4, A30, ', 2, ', I2, ']' ? ', A, A4, I2, /,
*         A, '[23;65H', ', ', A, '[23;59H', '$)

   IF (N.GT.3) WRITE(*,90) ESC, ESC, YELFORE, ALTERN,
*           M, ESC, WHITEFORE, OPCION, ESC, ESC
90 FORMAT (A, '[23;9H', 4X, A, A4, A30, ', . . . ', I2, ']' ? ', A, A4, 2X, I2, /,
*         A, '[23;65H', ', ', A, '[23;59H', '$)
95 READ (*,96) RESPUE
96 FORMAT (A1)

```

De acuerdo al valor leído se asigna un valor a I, para transformar el valor leído de tipo 'character' a tipo 'entero'.

```

IF (RESPUE.EQ.' ') I = OPCION
IF (RESPUE.EQ.'1') I = 1
IF (RESPUE.EQ.'2') I = 2
IF (RESPUE.EQ.'3') I = 3
IF (RESPUE.EQ.'4') I = 4
IF (RESPUE.EQ.'5') I = 5
IF (RESPUE.EQ.'6') I = 6
IF (RESPUE.EQ.'7') I = 7
IF (RESPUE.EQ.'8') I = 8
IF (RESPUE.EQ.'9') I = 9

```

Se verifica que I tenga un valor en el rango permitido de 1 a 10.

```
IF (I.EQ.10.OR.I.GT.N.OR.DEPEND(I).LT.0) GO TO 85
```

De acuerdo al tipo de variable asociada, se pasa a leer ésta.

```
IF (TIPVAL(I).NE.'CHA') GO TO 100
IF (VALCHA(I).NE.' ') GO TO 100
```

La opción seleccionada no tiene variable asociada y por tanto se sale de este subprograma.

```
OPCION = I
RETURN
```

Se limpia el espacio de pantalla donde se hace la pregunta acerca de la opción leída, la cual tiene una variable asociada.

```

100 WRITE (*, 110) ESC, ESC
110 FORMAT (A, '[23;10H', ', /, ')
*       A, '[23;40H', ', ')

```



```

* IF(L.GE.10) WRITE(*,50) ESC,L, ESC, WHITEFORE,I,ESC,YELFORE,
LISTOP(I), ESC, WHITEFORE, VALCHA(I)

```

La opción fué dada de baja del menú, por lo que se borra de pantalla.

```

128 L = KREGL * K + 4
    IF (L.GE.10) WRITE(*,129)ESC, L, ESC, L
129 FORMAT(A,'[',I2,';10H',',',/,
*         A,'[',I2,';30H',',',)
    IF (L.LT.10) WRITE(*,130)ESC, L, ESC, L
130 FORMAT(A,'[',I1,';10H',',',/,
*         A,'[',I1,';30H',',',)
135 CONTINUE
    GO TO 20
140 IF (TIPVAL(I).NE.'REA') GO TO 220
    VAR = VALNUM(I)
    IF (VAR.GE.0.0) GO TO 142

```

Se imprime una variable real menor que cero.

```

    VAR = ABS (VAR)
    IF(VAR.GE.1000.0)
*   WRITE (*,145) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, -VAR , ESC, ESC
    IF(VAR.GE.100.0.AND.VAR.LT.1000.0)
*   WRITE (*,146) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, -VAR , ESC, ESC
    IF(VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE (*,147) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, -VAR , ESC, ESC
    IF(VAR.GE.0.0.AND.VAR.LT.10.0)
*   WRITE (*,148) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, -VAR , ESC, ESC
    VAR = - VAR
    GO TO 155

```

Se imprime una variable real mayor que cero.

```

142 IF (VAR.LT.1.0)
*   WRITE (*,149) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, VAR, ESC, ESC
    IF (VAR.GE.1.0.AND.VAR.LT.10.0)
*   WRITE (*,150) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, VAR, ESC, ESC
    IF (VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE (*,151) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, VAR, ESC, ESC
    IF (VAR.GE.100.0.AND.VAR.LT.1000.0)
*   WRITE (*,152) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, VAR, ESC, ESC
    IF (VAR.GE.1000.0.AND.VAR.LT.10000.0)
*   WRITE (*,153) ESC, ESC, YELFORE, LISTOP(I), ESC,
*   WHITEFORE, VAR, ESC, ESC
145 FORMAT (A,'[23;10H',A,A4,'(',A44,'?',A,A4,F8.2,/,
*         A,'[23;68H',',',A,'[23;59H', '$)

```

```

146 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.2, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
147 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.3, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
148 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.4, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
149 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.5, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
150 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.5, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
151 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.4, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
152 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.3, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)
153 FORMAT (A, '[23;10H', A, A4, '(', A44, '? ', A, A4, F8.2, /,
*      A, '[23;68H', ' ', A, '[23;59H', '$)

```

Se lee la variable real VAR. Si su valor leído es cero, se asigna el valor inicial a la variable asociada con la opción. Si es distinto de cero, éste valor es el que se asigna a la variable asociada VALNUM(I).

```

155 VAR = 0.0
    READ(*,160) VAR
160 FORMAT(F10.5)
    IF (VAR.NE.0.0) VALNUM(I) = VAR

```

Se muestra el nuevo valor de la variable asociada a la opción seleccionada.

```

K = 1
DO 180 J = 1, N
  IF (DEPEND(J).LT.0) GO TO 180
  L = KRENGL * K + KPRIM + 1
  K = K + 1
  IF (J.EQ.I) GO TO 177
  GO TO 180
177  VAR = VALNUM(I)
     IF (VAR.GE.0.0) GO TO 178

```

Se imprime una variable real menor que cero.

```

VAR = ABS(VAR)
IF(L.LT.10.AND.VAR.GE.1000.0)
*      WRITE(*,51) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*      LISTOP(I), ESC, WHITEFORE, -VAR
IF(L.GE.10.AND.VAR.GE.1000.0)
*      WRITE(*,52) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*      LISTOP(I), ESC, WHITEFORE, -VAR
IF(L.LT.10.AND.VAR.GE.100.0.AND.VAR.LT.1000.0)
*      WRITE(*,53) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*      LISTOP(I), ESC, WHITEFORE, -VAR
IF(L.GE.10.AND.VAR.GE.100.0.AND.VAR.LT.1000.0)
*      WRITE(*,54) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*      LISTOP(I), ESC, WHITEFORE, -VAR

```

```

IF(L.LT.10.AND.VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE(*,55) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,-VAR
IF(L.GE.10.AND.VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE(*,56) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,-VAR
IF(L.LT.10.AND.VAR.GE.0.0.AND.VAR.LT.10.0)
*   WRITE(*,57) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,-VAR
IF(L.GE.10.AND.VAR.GE.0.0.AND.VAR.LT.10.0)
*   WRITE(*,58) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,-VAR

VAR = - VAR
GO TO 85

```

Se imprime una variable real mayor que cero.
--

```

178 IF(L.LT.10.AND.VAR.LT.1.0)
*   WRITE(*,59) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.GE.10.AND.VAR.LT.1.0)
*   WRITE(*,60) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.LT.10.AND.VAR.GE.1.0.AND.VAR.LT.10.0)
*   WRITE(*,61) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.GE.10.AND.VAR.GE.1.0.AND.VAR.LT.10.0)
*   WRITE(*,62) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.LT.10.AND.VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE(*,63) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.GE.10.AND.VAR.GE.10.0.AND.VAR.LT.100.0)
*   WRITE(*,64) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.LT.10.AND.VAR.GE.100.0.AND.VAR.LT.1000.0)
*   WRITE(*,65) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.GE.10.AND.VAR.GE.100.0.AND.VAR.LT.1000.0)
*   WRITE(*,66) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.LT.10.AND.VAR.GE.1000.0.AND.VAR.LT.10000.0)
*   WRITE(*,67) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR
IF(L.GE.10.AND.VAR.GE.1000.0.AND.VAR.LT.10000.0)
*   WRITE(*,68) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*   LISTOP(I),ESC,WHITEFORE,VAR

GO TO 85
180 CONTINUE
GO TO 85

```

Se pregunta por una variable entera asociada a la opción seleccionada.
--

```
220 IAUX = INT(VALNUM(I))
```



```

      IF(IAUX.LT.10) WRITE (*,230) ESC, ESC, YELFORE, LISTOP(I),
*           ESC, WHITEFORE, IAUX, ESC, ESC
      IF (IAUX.GE.10.AND. IAUX.LT.99)
*           WRITE (*,232) ESC, ESC, YELFORE, LISTOP(I),
*           ESC, WHITEFORE, IAUX, ESC, ESC
      IF (IAUX.GE.100.AND. IAUX.LT.999)
*           WRITE (*,234) ESC, ESC, YELFORE, LISTOP(I),
*           ESC, WHITEFORE, IAUX, ESC, ESC
230  FORMAT (A, '[23;10H', A, A4, ', 'Z', A44, '? ', A, A4, I1, /,
*           A, '[23;61H', ' ', A, '[23;59H', '$)
232  FORMAT (A, '[23;10H', A, A4, ', 'Z', A44, '? ', A, A4, I2, /,
*           A, '[23;62H', ' ', A, '[23;59H', '$)
234  FORMAT (A, '[23;10H', A, A4, ', 'Z', A44, '? ', A, A4, I3, /,
*           A, '[23;63H', ' ', A, '[23;59H', '$)
      I1 = 1

```

Se lee la variable entera asociada a la opción seleccionada.

```

      READ(*,240) I1
240  FORMAT(I4)

```

Se verifica que el valor de tal variable entera tenga un valor entre el intervalo mostrado en la siguiente instrucción.

```

      IF (I1.EQ.0.OR. I1.GT.999) GO TO 220

```

Se asigna a la variable real asociada a la opción la cantidad entera leída arriba.

```

      IAUX = I1
      IF (IAUX.NE.0) VALNUM(I) = REAL(IAUX)

```

Se imprime el nuevo valor de la variable entera asociada.

```

      K = 1
      DO 280 J = 1, N
          IF (DEPEND(J).LT.0) GO TO 280
          L = KRENGL * K + KPRIM + 1
          K = K + 1
          IF (J.EQ.1) GO TO 277
          GO TO 280
277  IF (L.GE.10) WRITE(*,129)ESC, L, ESC, L
      IF (L.LT.10) WRITE(*,130)ESC, L, ESC, L
      IF(L.LT.10.AND. IAUX.LT.10)
*           WRITE(*,70) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*           LISTOP(I), ESC, WHITEFORE, IAUX
      IF(L.GE.10.AND. IAUX.LT.10)
*           WRITE(*,71) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*           LISTOP(I), ESC, WHITEFORE, IAUX
      IF(L.LT.10.AND. IAUX.GE.10.AND. IAUX.LT.100)
*           WRITE(*,72) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*           LISTOP(I), ESC, WHITEFORE, IAUX
      IF(L.GE.10.AND. IAUX.GE.10.AND. IAUX.LT.100)
*           WRITE(*,73) ESC,L,ESC,WHITEFORE,I, ESC, YELFORE,
*           LISTOP(I), ESC, WHITEFORE, IAUX

```

```

IF(L.LT.10.AND.IAUX.GE.100.AND.IAUX.LT.1000)
*      WRITE(*,74) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*      LISTOP(I),ESC,WHITEFORE,IAUX
IF(L.GE.10.AND.IAUX.GE.100.AND.IAUX.LT.1000)
*      WRITE(*,75) ESC,L,ESC,WHITEFORE,I,ESC,YELFORE,
*      LISTOP(I),ESC,WHITEFORE,IAUX
280 CONTINUE

```

Se regresa a seleccionar otra opción.

```

GO TO 85
4926 FORMAT(A20)

```

Fin del subprograma menú.

```

RETURN
END

```

d) Subprograma CUADRO

Con este subprograma se pinta el marco azul que aparece en pantalla.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL CUADRO
```

y como se podrá notar, no contiene argumentos. Además, tampoco incluye bloques comunes. La descripción de variables utilizadas se da enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ESC	-	CHARACTER	Representa en Código ASCII a la tecla de 'escape'.
BLUBCK	-	CHARACTER	Al imprimirse el valor de esta variable, el efecto es un fondo azul en pantalla.
BLACK	-	CHARACTER	Al imprimirse el valor de esta variable, el efecto es caracteres negros.

El código de este subprograma es el siguiente:

Inicio del subprograma CUADRO.

```
SUBROUTINE    CUADRO
```

Declaración de variables.

```
CHARACTER  ESC*1, BLUBCK*4, BLACK*4
PARAMETER (BLUBCK = '[44m', BLACK = '[40m')
ESC = CHAR(27)
```

Se define como fondo para cualquier escritura el fondo azul.

```
WRITE (*,40) ESC, BLUBCK
40 FORMAT (A,A4)
```

Se limpia pantalla.

```
CALL CLEAR
```

Se pintan espacios blancos en negro, con el fondo azul.

```
DO 100 I = 2, 24
      IF (I.LT.10)
*      WRITE (*, 50) ESC, I, ESC, BLACK, ESC, I
50      FORMAT (A, '[', I1, ';3H', A, A4,
*              ,
*              , '/',
*              A, '[', I1, ';38H',
*              ,
*              ')
      IF (I.GE.10)
*      WRITE (*, 55) ESC, I, ESC, BLACK, ESC, I
55      FORMAT (A, '[', I2, ';3H', A, A4,
*              ,
*              , '/',
*              A, '[', I2, ';38H',
*              ,
*              ')
100 CONTINUE
```

Fin del subprograma cuadro.

```
RETURN
END
```

e) Subprograma INIMEN

Este subprograma tiene como función el inicializar las variables asociadas a un menú.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL  INIMEN
```

Las variables utilizadas en este subprograma son las incluidas en el bloque común MENU12, las cuales se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
VALNUM(15)	E/S	REAL	Se almacenan valores reales asociados con una de las opciones del menú.
DEPEND(15)	E/S	ENTERA	Define la dependencia entre dos opciones del menú, donde una de ellas aparecerá de acuerdo al valor que tome la otra.
LISTOP(15)	E/S	CHARACTER	Almacena el texto de cada una de las opciones mostradas en un menú.
ENCABE	E/S	CHARACTER	Contiene el encabezado que describe la menú actual.
ALTERN	E/S	CHARACTER	Contiene la pregunta acerca de la opción que se desea ejecutar.
TIPVAL(15)	E/S	CHARACTER	Contiene una cadena de tres caracteres para cada opción del menú, indicando el tipo de variable asociada.
VALCHA(15)	E/S	CHARACTER	Contiene una cadena de caracteres asociados a cada opción.

El código de este subprograma es el siguiente:

```
Inicio del subprograma INIMEN.
```

```
SUBROUTINE INIMEN
```

```
Declaración del bloque común MENU12.
```

```
COMMON /MENU12/ VALNUM, DEPEND, LISTOP, ENCABE, ALTERN, TIPVAL, VALCHA
```

```
Declaración de variables.
```

```
CHARACTER*50 LISTOP(15), ENCABE, ALTERN
CHARACTER*3 TIPVAL(15)
CHARACTER*12 VALCHA(15)
INTEGER DEPEND(15)
```

REAL*8

VALNUM(15)

Se inicializan variables y letreros asociados a cada opción: LISTOP(I) se llena de espacios en blanco, los cuales serán llenados en instrucciones posteriores del programa o subprograma que mandó ejecutar a INIMEN. Todas las opciones suponen una variable asociada del tipo 'character' al inicializar TIPVAL(I) = 'CHA'. El valor de la variable numérica asociada a cada opción se inicializa con 0.444444. En caso de que se requiera una variable tipo 'character' para cada opción, el valor asociado es ' '. Todas las opciones se consideran independientes entre sí, debido al valor dado a DEPEND(I).

```

DO 15 I = 1, 15
  LISTOP(I) = '
  TIPVAL(I) = 'CHA'
  VALNUM(I) = 0.444444
  VALCHA(I) = ' '
  DEPEND(I) = 0
15 CONTINUE

```

Fin del subprograma INIMEN.

```

RETURN
END

```

f) Subprograma VERIF

En este subprograma se verifican nombres de nodos introducidos por archivo o por interacción con el usuario, de acuerdo a la información de los nombres de nodos leídos en el primer bloque del archivo de datos.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL VERIF (NNODOS, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
```

Las variables involucradas en su argumento se describen enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
ITIP	E	ENTERA	Define el número de nodos a verificar. Si ITIP = 1, se revisa el nombre de dos nodos. Si ITIP = 2, se revisa el nombre de un nodo, únicamente.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
JX	E/S	ENTERA	Si el nombre del primer nodo revisado está en la lista DNOMB(.), se le asigna el número de la posición correspondiente en la lista. De otro modo, JX = 0.
JY	E/S	ENTERA	Si el nombre del segundo nodo revisado está en la lista DNOMB(.), se le asigna el número de la posición correspondiente en la lista. De otro modo, JX = 0.
I	E/S	ENTERA	Contador en cada bloque del archivo de datos que se está leyendo.
DNOMB(550)	E	CHARACTER	Lista de nombres de nodos, los cuales fueron registrados durante la lectura del primer bloque del archivo de datos.
NOMB1	E	CHARACTER	Nombre del primer nodo leído.
NOMB2	E	CHARACTER	Nombre del segundo nodo leído.
IEX	E/S	LOGICAL	Indicador del resultado de verificar los nombres de nodos.

El código de este subprograma se muestra a continuación:

```

C*****
C**** ESTE SUBPROGRAMA REvisa SI UN NODO O UN PAR DE NODOS LEIDOS - *
C**** SE ENCUENTRAN EN LA LISTA DE NODOS FORMADA - *
C**** EN LA PRIMERA PARTE DEL ARCHIVO DE DATOS. *
C *
C**** REALIZADO POR EL DR. RICARDO MOTA PALOMINO *****
C*****

```

Inicio del subprograma VERIF.

SUBROUTINE VERIF (NNODOS, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)

Declaración de variables.

```
CHARACTER*8      NOMB1, NOMB2, DNOMB(550)
CHARACTER*1      ESC
LOGICAL          IEX
```

Si los nombres de nodos a verificar están en DNOMB(.), IEX sale del subprograma con el valor asignado en la siguiente instrucción.

```
IEX= .TRUE.
ESC = CHAR(27)
```

Se verifica la existencia en DNOMB(.) del primer nombre de nodo.

```
DO 110 J = 1, NNODOS
      IF (DNOMB(J).NE.NOMB1) GO TO 110
```

Se ha encontrado el primer nombre en la lista.

```
JX = J
GO TO 112
110 CONTINUE
```

Se imprime mensaje de la no existencia del primer nombre.

```
CALL CUADRO
WRITE (*, 111) ESC, NOMB1
111 FORMAT(A, '[19;21H', ' ** EL NODO ', A8, ' NO EXISTE ')
CALL ENCA
```

El contador I del proceso de lectura de datos se reduce en 1.

```
I = I - 1
```

IEX se actualiza con .FALSE. indicando que el primer nombre no existe.

```
IEX = .FALSE.
```

JX = 0 para indicar lo mismo que IEX = .FALSE.

```
JX = 0
```

Se revisa si sólo un nombre de nodo se está verificando (ITIP = 2).

```
112 IF (ITIP.EQ.2) RETURN
```

Se verifica el segundo nombre de nodo.

```
DO 116 J = 1, NNODOS
      IF (DNOMB(J).NE.NOMB2) GO TO 116
```

Se ha encontrado el segundo nombre de nodo en la lista DNOMB(.).

```
JY = J
```

Se sale de este subprograma.

```
RETURN
```

116 CONTINUE

No se encuentra el segundo nombre de nodo en la lista y se envía el mensaje correspondiente.

```
CALL CUADRO
WRITE (*, 111) ESC, NOMB2
```

El contador I del proceso de lectura de datos se reduce en 1.

```
I = I - 1
```

IEX se actualiza con .FALSE. indicando que el segundo nombre no existe.

```
IEX = .FALSE.
CALL ENCA
```

JY = 0 para indicar lo mismo que IEX = .FALSE.

```
JY = 0
```

Fin del subprograma VERIF.

```
118 RETURN
END
```

g) Subprograma SINO

Este subprograma tiene como función el regresar los valores 'SI' o 'NO', cuando se le pregunta al usuario por una de estas opciones.

Para acceder este subprograma se usa la instrucción:

```
CALL SINO (RESPUE, ANSWER)
```

Las variables usadas en el argumento se describen enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ANSWER	E/S	CHARACTER	Esta variable se inicializa en el subprograma que llamó a SINO.
RESPUE	E/S	CHARACTER	De acuerdo al valor de la variable ANSWER, RESPUE se regresa con el valor de 'SI' o 'NO'.

El código correspondiente a este subprograma es el siguiente:

Inicio del subprograma SINO.

```
SUBROUTINE SINO (RESPUE, ANSWER)
```

Declaración de variables.

```
CHARACTER*2  RESPUE, ANSWER
```

C

```
IF (ANSWER.EQ.'SI') GO TO 10
```

El valor de RESPUE se actualiza a 'NO' al cumplirse esta condición.

```
IF (RESPUE.EQ.'NO'.OR.RESPUE.EQ.'No'.OR.RESPUE.EQ.'no'.OR.
*      RESPUE.EQ.'no'.OR.RESPUE.EQ.' ')RESPUE='NO'
```

El valor de RESPUE se actualiza a 'SI' al cumplirse esta condición.

```
IF (RESPUE.EQ.'SI'.OR.RESPUE.EQ.'Si'.OR.RESPUE.EQ.'sI'.OR.
*      RESPUE.EQ.'si')      RESPUE='SI'
RETURN
```

El valor de RESPUE se actualiza a 'SI' al cumplirse esta condición.

```
10 IF (RESPUE.EQ.'SI'.OR.RESPUE.EQ.'Si'.OR.RESPUE.EQ.'sI'.OR.
*      RESPUE.EQ.'si'.OR.RESPUE.EQ.' ')RESPUE='SI'
```

El valor de RESPUE se actualiza a 'NO' al cumplirse esta condición.

```
IF (RESPUE.EQ.'NO'.OR.RESPUE.EQ.'No'.OR.RESPUE.EQ.'no'.OR.
*      RESPUE.EQ.'no')      RESPUE='NO'
```

Fin del subprograma SINO.

```
RETURN
END
```

h) Subprograma SALIAS

Este subprograma tiene la función de verificar la opción de terminar una sesión de simulación, seleccionada por el usuario.

La ejecución de este subprograma se logra mediante la instrucción:

```
CALL SALIAS (INDSAL)
```

La descripción de variables involucradas en este subprograma se muestra a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
INDSAL	E/S	ENTERA	Indicador de que la sesión habrá de terminar (INDSAL = 0), o continuará (INDSAL = 1).
ANSWER	E/S	CHARACTER	Esta variable se inicializa y envía a SINO para definir el valor de RESPUE.
RESPUE	E/S	CHARACTER	Esta variable se lee y se envía a SINO para regresar con el valor de 'SI' o 'NO'.
YELFORE	-	CHARACTER	Pinta caracteres en color marrón.
WHITEFORE	-	CHARACTER	Pinta caracteres en color blanco.

El código correspondiente a este subprograma se muestra enseguida.

Inicio del subprograma SALIAS.

```
SUBROUTINE SALIAS (INDSAL)
```

Declaración de variables.

```
CHARACTER ANSWER*2, RESPUE*2, ESC*1, YELFORE*4, WHITEFORE*4
PARAMETER (YELFORE = '[33m', WHITEFORE = '[37m')
ESC = CHAR(27)
```

Se pinta el marco azul.

```
CALL CUADRO
```

Se inicializan INDSAL y ANSWER tal que se termine la simulación.

```
INDSAL = 0
10 ANSWER = 'SI'
```

Se inicializa RESPUE con un espacio en blanco, y se lee enseguida, con la indicación de que el valor de 'default' es 'SI'.

```
RESPUE = ' '
WRITE (*, 15) ESC, YELFORE, ESC, ESC, WHITEFORE, ESC, ANSWER, ESC
15 FORMAT(A, A4, A, '[10;10H', '¿ REALMENTE DESEA SALIR DEL PROGRAMA',
*      ' [SI, NO] ? ', /, A, A4, A, '[10;61H', A2, A, '[10;61H', $)
READ (*, 16) RESPUE
16 FORMAT (A2)
CALL SINO (RESPUE, ANSWER)
```

Se verifica el valor de RESPUE.

```
IF (RESPUE.NE.'SI'.AND.RESPUE.NE.'NO') GO TO 10
```

Si se cumple esta condición, la sesión de simulación continuará.

```
IF (RESPUE.EQ.'NO') INDSAL = 1
```

Fin del subprograma SALIAS.

```
RETURN
END
```

i) Subprograma CICLO

Este subprograma tiene el propósito de crear un retardo de tiempo en la presentación de la portada del simulador mediante la ejecución de dos ciclos iterativos, cuyos límites se definen en el subprograma PANTAL.

Este subprograma se invoca mediante la instrucción:

```
CALL CICLO (N1, N2)
```

cuyos argumentos se describen enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
N1	E	ENTERA	Límite superior del ciclo iterativo exterior.
N2	E	ENTERA	Límite superior del ciclo iterativo interior.

El código de este subprograma es el siguiente:

Inicio del subprograma CICLO.

```
SUBROUTINE CICLO (N1, N2)
DO 2 I = 1, N1
DO 1 J = 1, N2
1 CONTINUE
2 CONTINUE
```

Fin del subprograma CICLO.

```
RETURN
END
```

7. Subprogramas Algorítmicos

Este conjunto de subprogramas representa al algoritmo de flujos de potencia, involucrando actividades como la creación de circuitos Π de elementos de transmisión, selección del nodo compensador, determinación de islas eléctricas, construcción de la matriz de admitancias nodal y el proceso iterativo utilizando el método Desacoplado Rápido. Tales subprogramas se describen a continuación.

a) Subprograma GRAFIC

La función de este subprograma es la determinación del número de islas eléctricas en el sistema. Cada isla se representa por un gráfica conectada no dirigida, la cual se almacena en forma de una lista encadenada.

Este subprograma es accesado mediante la instrucción:

```
CALL  GRAFIC  (NNODOS, NISLAS, INISLA, ISLA, NODFUE, KFUERA)
```

Una descripción de las variables involucradas en este subprograma se muestra enseguida.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NISLAS	E/S	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E/S	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E/S	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
NODFUE(KFUERA)	E	ENTERA	Lista de nodos que han sido dados de baja del sistema, ya sea por algoritmo o por decisión del usuario.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
KFUERA	E	ENTERA	Número de nodos que han sido dados de baja del sistema.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
INICPR(NNODOS)	E/S	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.
IFINPR(2*NLINES)	E/S	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NE(i) indica el número del nodo de envío.
NR(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i.

El código de este subprograma es el siguiente:

Inicio del subprograma GRAFIC.

SUBROUTINE GRAFIC (NNODOS, NISLAS, INISLA, ISLA, NODFUE, KFUERA)

C

Declaración del bloque común CIRCUI y de variables.

COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR

C

INTEGER INISLA(10), ISLA(550), INICPR(550), IFINPR(2000),
* IPOSPR(2000), NE(1000), NR(1000), NODFUE(15)

Inicialización de la variable NISLAS.

```
NISLAS = 1
DO 5 I = 1, NNODOS
```

Se verifica que el nodo I no esté dado de baja.

```
DO K1= 1, KFUERA
```

Al cumplirse esta condición se detecta que el nodo I está dado de baja.

```
IF (NODFUE(K1).EQ.I) GO TO 5
END DO
```

I no está dado de baja, por tanto se inicia ISLA(.) con este nodo.

```
ISLA(1) = I
GO TO 7
5 CONTINUE
```

La primera isla se inicia en la primera posición de ISLA, y por ésto, INISLA(1) = 1.

```
7 INISLA(1) = 1
```

INICIA es el límite inferior del ciclo iterativo 30 y se inicializa con el valor de 1.

```
INICIA = 1
```

K1 es un contador de los nodos visitados en la gráfica parcial.
K2 es el contador de los nodos agregados a la gráfica parcial.
K3 es el nodo actual que se visita en la gráfica.
NINT es el número de nodos dados de alta en el sistema eléctrico.

```
K1 = 0
K2 = 1
K3 = 1
NINT = NNODOS - KFUERA
```

Si esta condición se cumple en algún momento significa que no existen más nodos que agregar a la gráfica parcial.

```
10 IF(K1.EQ.K2) GO TO 50
```

Si esta condición se cumple significa que se han revisado todos los nodos del sistema eléctrico.

```
15 IF(K2.EQ.NINT) GO TO 80
```

I es el nodo actual que se visita de la gráfica parcial. K indica la posición inicial de la parte de la lista encadenada de circuitos π que corresponden al nodo I.

```
I = ISLA(K3)
K = INICPR(I)
```

Se barre la información del nodo I.

```
17      L = IPOSPR(K)
        LI = NE(L)
```

Se verifica que el elemento de transmisión esté dado de alta.

```
IF (LI.LT.0) GO TO 40
```

El elemento de transmisión está dado de alta y ahora se verifica que LI sea el nodo del elemento de transmisión distinto al nodo I.

```
IF(I.EQ.NR(L)) GO TO 20
LI = NR(L)
```

Se verifica que el nodo LI, que se agregará a la lista de la subgráfica esté dado de alta en el sistema.

```
DO J = 1, KFUERA
  IF (NODFUE(J).EQ.LI) GO TO 40
END DO
```

Se verifica que LI no esté incluido en la gráfica parcial.

```
20      DO 30 M = INICIA, K2
        IF (LI.EQ.ISLA(M)) GO TO 40
30      CONTINUE
```

LI no está incluido en la gráfica parcial y ahora se incluye.

```
K2 = K2 + 1
ISLA(K2) = LI
```

Se busca otra posición relacionada con el nodo I.

```
40      K = IFINPR(K)
        IF (K.EQ.0) GO TO 45
        GO TO 17
```

Se han revisado todos los elementos de transmisión relacionados con I. Se incrementa K1 en 1 para indicar que se ha visitado un nodo más. Se incrementa K3 en 1 para visitar el siguiente nodo de la gráfica.

```
45 K1 = K1 + 1
    K3 = K1 + 1
    GO TO 10
```

Se ha encontrado que existen dos o más islas. Aquí, se actualiza el número de las mismas y se apunta en INISLA(NISLAS) la posición de la lista encadenada donde se inicia la información de la nueva isla.

```
50 NISLAS = NISLAS + 1
    INISLA(NISLAS) = K2 + 1
```

Se busca un nodo para empezar a crear la gráfica de la nueva isla.

```
DO 70 J = 2, NNODOS
```

```

DO 65 JJ = 1, K2
      IF(J.EQ.ISLA(JJ)) GO TO 70
65  CONTINUE
DO  K4 = 1, KFUERA
      IF (J.EQ.NODFUE(K4)) GO TO 70
END DO

```

Al llegar a este punto, el nodo J no está incluido en las islas creadas anteriormente (ciclo iterativo 65) y está dado de alta (ciclo iterativo anterior), por lo que es el nodo con el que se inicia la gráfica de la nueva isla.

```

      ISLA(K1 + 1) = J
      K2 = K2 + 1
      INICIA = K2
      GO TO 15
70 CONTINUE

```

Fin del subprograma GRAFIC.

```

80 RETURN
END

```

b) Subprograma INICFC

La función de este subprograma es la inicialización de factores por los cuales se afecta la carga en una isla.

La instrucción mediante la cual se ejecuta este subprograma es:

```
CALL INICFC (NISLAS, FPCISLA, FQCISLA)
```

cuyos argumentos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
FPCISLA(NISLAS)	E/S	REAL	FPCISLA(I) tiene el factor para modificar la potencia real de carga en la isla I.
FQCISLA(NISLAS)	E/S	REAL	FQCISLA(I) tiene el factor para modificar la potencia reactiva de carga en la isla I.

El código de este subprograma es el siguiente:

Inicio del subprograma INICFC.

```
SUBROUTINE INICFC (NISLAS, FPCISLA, FQCISLA)
```

Declaración de variables.

```
DIMENSION FPCISLA(10), FQCISLA(10)
```

Los factores se inicializan en 1.0

```
DO I = 1, NISLAS
  FPCISLA(I) = 1.0
  FQCISLA(I) = 1.0
END DO
```

Fin del subprograma INICFC.

```
RETURN
END
```

c) Subprograma CIRCUI

Este subprograma tiene por objetivo calcular circuitos π de elementos de transmisión del sistema eléctrico en función de admitancias.

La instrucción para ejecutar este subprograma es:

```
CALL CIRCUI (YPQ2E, YPQ2R, R, X, R1, X1, GPQ2E, GPQ2R)
```

cuyos argumentos y bloques comunes incluidos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
YPQ2E(NLINES)	E/S	REAL	Mitad del efecto capacitivo asignado al nodo de envío en el circuito π de cada elemento de transmisión.
YPQ2R(NLINES)	S	REAL	Mitad del efecto capacitivo asignado al nodo de recepción en el circuito π de cada elemento de transmisión.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
R(NLINES)	E	REAL	Resistencia serie del circuito π de cada elemento de transmisión.
X(NLINES)	E	REAL	Reactancia inductiva serie del circuito π de cada elemento de transmisión.
R1(NLINES)	S	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	S	REAL	Susceptancia inductiva serie del circuito π de cada elemento de transmisión.
GPQ2E(NLINES)	S	REAL	Parte real del efecto en derivación asignado al nodo de envío en el circuito π de cada elemento de transmisión.
GPQ2R(NLINES)	S	REAL	Parte real del efecto en derivación asignado al nodo de recepción en el circuito π de cada elemento de transmisión.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NE(i) indica el número del nodo de envío.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NR(NLINES)	E	ENTERA	Para el circuito π i -ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i .
NTAPS	E	ENTERA	Total de transformadores con tap variable.
NETR(NTAPS)	E	ENTERA	Identificador del nodo al lado del tap de transformadores con tap variable.
NRTR(NTAPS)	E	ENTERA	Identificador del nodo al lado contrario del tap de transformadores con tap variable.
TAP(NTAPS)	E	REAL	Valor del tap en cada uno de los transformadores con tap variable.

El código de este subprograma se muestra a continuación.

Inicio del subprograma CIRCUI.

```
SUBROUTINE CIRCUI (YPQ2E, YPQ2R, R, X, R1, X1, GPQ2E, GPQ2R)
```

Declaración de bloques comunes y de variables.

```
COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
COMMON /TRANSF/ NTAPS, NETR, NRTR, TAP
```

```
INTEGER NE(1000), NR(1000), NETR(300), NRTR(300), INICPR(550),
* IFINPR(2000), IPOSPR(2000)
```

```
DIMENSION YPQ2E(1000), YPQ2R(1000), R(1000), X(1000),
* TAP(300), GPQ2E(1000), R1(1000), X1(1000), GPQ2R(1000)
```

Se construyen los circuitos π de cada elemento de transmisión.

```
DO 30 I = 1, NLINES
```

Para líneas $YPQ2E(I) = YPQ2R(I)$. Para transformadores $YPQ2E(I) \neq YPQ2R(I)$ y si son distintas estas variables se deduce que ya se ha calculado el circuito π de un transformador.

```
IF (YPQ2E(I).NE.YPQ2R(I)) GO TO 30
```

Se calcula conductancia $-R1(I)-$ y susceptancia inductiva serie $-X1(I)-$ del circuito π .

```
15      G = R(I)
        B = X(I)
        T = G * G + B * B
        G = G / T
        B = - B / T
        R1(I) = G
        X1(I) = B
```

Se investiga si el elemento I es un transformador.

```
DO 20 J = 1, NTAPS
      IF (NE(I).NE.NETR(J).OR.NR(I).NE.NRTR(J)) GO TO 20
```

El elemento I es un transformador. Se calcula el circuito π del mismo.

```
T = TAP(J)
```

Esta condición se cumple cuando se está visitando un transformador al cual ya se le ha calculado su circuito π .

```
IF (TAP(J).LT.0.0) GO TO 20
YPQ2E(I) = B * (1.0 - T) / (T * T)
YPQ2R(I) = B * (T - 1.0) / T
R1(I) = G / T
X1(I) = B / T
```

Se cambia el signo al valor del tap de este transformador para indicar que ya ha sido visitado. Esto resulta útil en caso de transformadores en paralelo.

```
TAP(J) = - T
IF (G.EQ.0.0) GO TO 30
GPQ2E(I) = G * (1.0 - T) / (T * T)
GPQ2R(I) = G * (T - 1.0) / T
GO TO 30
```

```
20      CONTINUE
30      CONTINUE
```

Se restablece el valor original de cada transformador.

```
DO 35 I = 1, NTAPS
35      TAP(I) = ABS ( TAP(I) )
```

Fin del subprograma CIRCUI.

```
RETURN
END
```

d) Subprograma YNODAL

Con este subprograma se construye la matriz de admitancias nodal, cuya información se compacta usando el segundo esquema de empaquetamiento y listas encadenadas. Esta matriz se forma por inspección.

La manera de invocarlo es mediante la instrucción:

```
CALL  YNODAL  (R1,X1, YPQ2E, YPQ2R, SHUNT, NNODOS, NSHNTS, NSHUNT,
*          INIC, ICOL, YBUSR, YBUSI, ISP, NLT, IERR, GPQ2E, GPQ2R,
*          LONG, KFUERA, NODFUE)
```

La descripción de cada uno de sus argumentos y bloques comunes asociados se muestra a continuación:

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
R1(NLINES)	E	REAL	Conductancia serie del circuito π de cada elemento de transmisión.
X1(NLINES)	E	REAL	Susceptancia inductiva serie del circuito π de cada elemento de transmisión.
GPQ2E(NLINES)	E	REAL	Parte real del efecto en derivación asignado al nodo de envío en el circuito π de cada elemento de transmisión.
GPQ2R(NLINES)	E	REAL	Parte real del efecto en derivación asignado al nodo de recepción en el circuito π de cada elemento de transmisión.
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NSHNTS	E	ENTERA	Número total de elementos de compensación fija en derivación.
NSHUNT(NSHNTS)	E	ENTERA	Identificador de cada nodo en el cual existe un compensador en derivación.
INIC(NNODOS)	S	ENTERA	Indica la posición en la que inicia la información de cada nodo de la matriz de admitancias nodal en un segundo esquema de empaquetamiento.
LONG	S	ENTERA	$LONG = NNODOS + 2 * NLINES.$
ICOL(LONG)	S	ENTERA	ICOL(k) indica la columna donde se encuentra un elemento de la matriz de admitancias nodal distinto de cero, asociado al renglón i, donde ICOL(k) se puede acceder barriendo la información encadenada a partir de INIC(i).
YBUSR(LONG)	S	REAL	Parte real de un elemento de la matriz de admitancias nodal.
YBUSI(LONG)	S	REAL	Parte imaginaria de un elemento de la matriz de admitancias nodal.
ISP(LONG)	S	ENTERA	Indica la siguiente posición en la información empaquetada donde se encuentra otro elemento del mismo renglón.
NLT(NNODOS)	S	ENTERA	Si $NLT(i)=1$, indica que el nodo i tiene compensación fija en derivación conectada. Si $NLT(i) = 0$, no existe compensación fija en derivación conectada en el nodo i.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i , iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i -ésimo, NE(i) indica el número del nodo de envío.
NR(NLINES)	E	ENTERA	Para el circuito π i -ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i .
IERR	E/S	ENTERA	IERR = 0 indica que la matriz de admitancias nodal se construyó en forma satisfactoria. IERR = 1 indicará que existió error en la construcción de la matriz de admitancias nodal.

El código de este subprograma se describe a continuación.

Inicio del subprograma YNODAL.

```

C
C   SUBROUTINA PARA FORMAR LA MATRIZ DE ADMITANCIAS NODAL,
C   UTILIZANDO TECNICAS DE DISPERSIDAD.
C
C   SUBROUTINE YNODAL (R1, X1, YPQ2E, YPQ2R, SHUNT, NNODOS, NSHNTS, NSHUNT,
*                   INIC, ICOL, YBUSR, YBUSI, ISP, NLT, IERR, GPQ2E,
*                   GPQ2R, LONG, KFUERA, NODFUE)

```

Declaración de bloques comunes y de variables.

```

COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
C
DIMENSION R1(1000), X1(1000), YPQ2E(1000), YPQ2R(1000), YBUSR(2550),
*         NR(1000), GPQ2E(1000), NLT(550), IPOSPR(2000), SHUNT(150),
*         ICOL(2550), NE(1000), YBUSI(2550), INIC(550), NSHUNT(150),
*         INICPR(550), GPQ2R(1000), ISP(2550), IFINPR(2000), NODFUE(15)

```

Se inicializan los arreglos asociados a la matriz de admitancias nodal.

```

DO 1 I = 1, NNODOS
    NLT(I) = 0
1    INIC(I) = 0
    LONG = NNODOS + 2 * NLINES
DO 2 I = 1, LONG
    ICOL(I) = 0
    YBUSR(I) = 0.0
    YBUSI(I) = 0.0
    ISP(I) = 0
2 CONTINUE

```

IERR se inicializa considerando que no existirá error al construir la matriz de admitancias nodal. K es el contador para definir la posición donde se localiza el elemento diagonal del renglón I en la misma matriz

```

IERR = 0
K = 1

```

Se inicia construcción.

```

DO 300 I = 1, NNODOS
    INIC(I) = K

```

ICOL(K) contendrá el número de elementos distintos de cero del renglón I, por lo cual se inicializa con valor de 0.

```

ICOL(K) = 0

```

Se revisa si el nodo I está dado de alta.

```

DO 10 J = 1, KFUERA
    IF (NODFUE(J).NE.I) GO TO 10

```

EL nodo I está dado de baja.

```

    K = K + 1
    GO TO 300
10 CONTINUE

```

El nodo I está dado de alta. K2 es un contador de elementos fuera de la diagonal en el renglón I. Se inicia la búsqueda de elementos de transmisión asociados al nodo I.

```

    K2 = 1
    KC = INICPR(I)
15    J = IPOSPR(KC)

```

Se verifica que el elemento de transmisión esté dado de alta.

```

    IF (NE(J).LT.0) GO TO 200
    IF (I.EQ.NE(J)) GO TO 20
    IF (I.NE.NR(J)) GO TO 200

```


El nodo I es el nodo de recepción del elemento de transmisión J. NAUX contendrá al otro nodo extremo.

```

NAUX = NE(J)
YPQ2 = YPQ2R(J)
GPQ2 = GPQ2R(J)
GO TO 22

```

El nodo I es el nodo de envío del elemento de transmisión J. NAUX contendrá al otro nodo extremo.

```

20      NAUX = NR(J)
        YPQ2 = YPQ2E(J)
        GPQ2 = GPQ2E(J)

```

K1 es un contador de la posición actual que se está llenando en los arreglos representando a la matriz de admitancias nodal.

```

22      K1 = K + K2
        GPQ = - R1(J)
        BPQ = - X1(J)
        IF(ICOL(K).EQ.0) GO TO 50

```

Se verifica que NAUX no haya sido incluido anteriormente. Esto ocurre en casos donde existen elementos de transmisión en paralelo.

```

DO 25 L = K + 1, K1 - 1
      IF(NAUX.NE.ICOL(L)) GO TO 25

```

NAUX ya estaba apuntado en ICOL(.) y por tanto, lo único que se hace es modificar el elemento de la matriz correspondiente para incluir el elemento de transmisión en paralelo.

```

        K1 = K1 - 1
        YBUSR(L) = YBUSR(L) + GPQ
        YBUSI(L) = YBUSI(L) + BPQ
        GO TO 60
25      CONTINUE

```

NAUX no estaba incluido, por lo que se genera un nuevo elemento de la matriz de admitancias nodal.

```

50      ICOL(K1) = NAUX
        ICOL(K) = ICOL(K) + 1
        YBUSR(K1) = GPQ
        YBUSI(K1) = BPQ
        ISP(K1 - 1) = K1
        K2 = K2 + 1

```

Se actualiza el valor del elemento diagonal del renglón I. de la matriz

```

60      YBUSR(K) = YBUSR(K) - GPQ - GPQ2
        YBUSI(K) = YBUSI(K) - BPQ + YPQ2

```

Se selecciona otro elemento de transmisión asociado al nodo I.

```

200          KC = IFINPR(KC)
           IF (KC.EQ.0) GO TO 205

```

Existe otro elemento de transmisión asociado al nodo I, por lo que se regresa a repetir el proceso.

```
GO TO 15
```

No existe otro elemento de transmisión asociado al nodo I, por lo que se llena la posición actual de ISP con 0, para indicar que termina la lista encadenada del renglón I. Además, K se actualiza para iniciar la localización del siguiente renglón.

```

205          ISP(K1) = 0
           K = K1 + 1

```

```
300 CONTINUE
```

C

C SE MODIFICA YBUS AL INCLUIR COMPENSADORES

C

Si no existe compensación en derivación se va directamente a la salida del subprograma.

```
400 IF(NSHNTS.EQ.0)GO TO 520
```

El nodo K tiene compensación en derivación y se modifica la matriz de admitancias nodal en la parte imaginaria de su elemento diagonal.

```

DO 330 I = 1, NSHNTS
           K = NSHUNT(I)
           IF (K.LT.0) GO TO 330
           NLT(K) = 1
           K = INIC(K)
           YBUSI(K) = YBUSI(K) + SHUNT(I)

```

```

330 CONTINUE
GO TO 520

```

Hay algún error en la información. Se envía el mensaje correspondiente. Además, se cambia el valor de IERR a 1, para que el programa principal detecte el error.

```

500 WRITE(6,510)K,K1
510 FORMAT(//10X,'ENTRE LOS NODOS ',I3,' Y ',I3,' NO HAY',
*          ' CONEXION ALGUNA. REVISE LOS DATOS !!!',//)
IERR = 1

```

Fin del subprograma YNODAL.

```

520 CONTINUE
RETURN
END

```

e) Subprograma ASIGNA

En este subprograma se selecciona el nodo compensador de cada isla eléctrica que exista en el sistema. Esta selección puede efectuarse en forma automática o por decisión del usuario.

La instrucción para invocarlo es la siguiente:

```
CALL ASIGNA (NISLAS, ISLA, INISLA, PG, TIPO, ISLACK, NNODOS,
*          NNGEN, DNOMB, NPV, PSLACK)
```

Cada argumento se describe a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
PG(NNODOS)	E	REAL	Potencia real de generación en cada nodo del sistema eléctrico de potencia.
TIPO(NNODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
ISLACK(NISLAS)	S	ENTERA	ISLACK(i) contiene al nodo compensador de la isla i.
PSLACK(NISLAS)	S	ENTERA	Potencia real nominal de generación del nodo compensador de cada isla.
NNGEN	S	ENTERA	Número total de nodos de generación en el sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
DNOMB(NNODOS)	E	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.

El código correspondiente a este subprograma se muestra a continuación.

Inicio del subprograma ASIGNA.

```
SUBROUTINE ASIGNA (NISLAS, ISLA, INISLA, PG, TIPO, ISLACK, NNODOS,
* NNGEN, DNOMB, NPV, PSLACK)
```

Declaración de variables.

```
REAL PG(550), PSLACK(10)
C
INTEGER TIPO(550), NPV(150), INISLA(10), ISLACK(10), ISLA(550)
C
CHARACTER NOMBRE*8, NOMB*2, ESC*1, WHITEFORE*4, YELFORE*4
CHARACTER*8 DNOMB(550)
PARAMETER (YELFORE='[33m', WHITEFORE = '[37m')
C
ESC = CHAR(27)
```

Se selecciona el compensador de cada isla.

```
DO 90 I = 1, NISLAS
```

K es el límite inferior del ciclo iterativo para acceder la lista de nodos de la isla actual, almacenada en localidades contiguas de memoria
K2 es el límite superior de este ciclo.

```
K = INISLA(I)
IF (I.LT.NISLAS) K2 = INISLA(I + 1) - 1
IF (I.EQ.NISLAS) K2 = NNODOS
```

SE limpia pantalla y se pinta el marco azul.

```
CALL CUADRO
```

Se pregunta al usuario si desea seleccionar el nodo compensador de la isla actual I. La respuesta de 'default' es NO.

```
5204 NOMB = 'NO'
WRITE (*, 5205) ESC, ESC, YELFORE, I
WRITE (*, 5206) ESC, ESC, WHITEFORE, NOMB, ESC
5205 FORMAT (A, '[15;4H', A, A4,
* '¿ DESEA SELECCIONAR EL NODO COMPENSADOR DE LA ISLA ',
* I2, ' [SI, NO] ? ')
```

```

5206      FORMAT (A, '[15;71H', A, A4, A2, A, '[15;71H', $)
          READ (*, 357, ERR=401) NOMB
357       FORMAT (A2)
          IF (NOMB.EQ. 'SI'.OR.NOMB.EQ. 'si') GO TO 400
          IF (NOMB.EQ. 'NO'.OR.NOMB.EQ. 'no'.OR.NOMB.EQ. ' ')GO TO 491

```

La respuesta no es aceptable, por lo que se regresa a efectuar la misma pregunta.

```
401       GO TO 5204
```

La respuesta es afirmativa, por lo que se procede a preguntar ahora por el nombre del nodo que será el compensador.

```

400       WRITE (*, 5212)ESC
5212      FORMAT (A, '[17;9H', '
*
410       WRITE (*, 5215) ESC
          WRITE (*, 5216) ESC
5215      FORMAT (A, '[17;9H',
*
          '¿ NOMBRE DEL NODO [maximo 8 caracteres] ?'
*
          ')
5216      FORMAT (A, '[17;54H', $)
          READ(*, 447, ERR=401) NOMBRE
          WRITE (*, 457) ESC
447       FORMAT (A8)

```

Se revisa que el nombre del nodo esté dado de alta en DNOMB(.).

```

          DO 450 J = 1, NNODOS
          IF (DNOMB(J).EQ.NOMBRE) GO TO 460
450       CONTINUE

```

El nombre dado por el usuario es incorrecto. Se imprime mensaje.

```

          WRITE (*, 455) ESC, NOMBRE
455       FORMAT (A, '[22;10H', '**** EL NODO ', A8,
*
          ' NO EXISTE EN EL SISTEMA ')
457       FORMAT(A, '[22;10H', '
*
          ')
          CALL ENCA
          CALL CUADRO

```

Se regresa a preguntar nuevamente el nombre del nodo.

```
GO TO 410
```

El nombre del nodo es correcto. Ahora, se verifica que corresponda a un nodo de generación.

```

460       DO 470 L = 1, NNGEN
          IP = NPV(L)
          IF (IP.EQ.J) GO TO 480
470       CONTINUE

```

El nodo seleccionado no es de generación.

```

475      WRITE (*,475) ESC, NOMBRE
*      FORMAT (A,'[22;10H','***** EL NODO ',A8,
          ' NO ES DE GENERACION ... SELECCIONE OTRO ')
      CALL ENCA
      CALL CUADRO

```

Se regresa a preguntar por el nombre de otro nodo.

GO TO 410

Ahora, se verifica que el nodo seleccionado está en la isla actual.

```

480      DO 482 L = K, K2
          IF (ISLA(L).EQ.IP) GO TO 485
482      CONTINUE

```

El nodo seleccionado no pertenece a la isla actual. Se imprime mensaje.

```

483      WRITE (*,483) ESC, NOMBRE
*      FORMAT (A,'[22;10H','***** EL NODO ',A8,
          ' NO PERTENECE A ESTA ISLA')
      CALL ENCA
      CALL CUADRO

```

Se regresa a preguntar por el nombre de otro nodo.

GO TO 410

El nodo seleccionado por el usuario será el compensador. Se hacen las especificaciones necesarias.

```

485      CONTINUE
          ISLACK(I) = IP
          TIPO(IP) = 1
          PSLACK(I) = PG(IP)

```

Se va a otra isla.

GO TO 90

C

C SE SELECCIONA AUTOMATICAMENTE EL COMPENSADOR DE LA ISLA ACTUAL

C

Se ha optado por seleccionar automáticamente el nodo compensador en cada isla. Para ésto se selecciona el nodo con mayor potencia real de generación de cada isla.

```

491      PMAX = -0.0001
          DO 80 L = K, K2
              NK = ISLA(L)
              IF (NK.LT.0) GO TO 80
              IF (TIPO(NK).EQ.3) GO TO 80
              IF (PMAX.GE.PG(NK)) GO TO 80
              PMAX = PG(NK)

```

```

                NODO = NK
80          CONTINUE

```

Se verifica que la isla actual tenga al menos un nodo con potencia real de generación mayor que cero.

```

                IF (PMAX.GT.0.0) GO TO 88

```

La isla actual no tiene generación de potencia real.

```

                WRITE (*, 85) ESC, I
85          FORMAT (A, '[20;20H', 'LA ISLA ', I2, ' NO TIENE GENERACION')

```

Se elimina la isla del proceso de solución mediante un cambio de signo en INISLA(I).

```

                INISLA(I) = - K
                CALL ENCA

```

Se va a otra isla.

```

                GO TO 90

```

Se efectúa la asignación del nodo compensador en la isla actual.

```

88          ISLACK(I) = NODO
                TIPO(NODO) = 1
                PSLACK(I) = PG(NODO)
90 CONTINUE

```

Se imprimen los nodos que se han seleccionado como compensadores para cada isla.

```

                CALL CUADRO
                K2 = 2
                DO 99 I = 1, NISLAS
                    IF (INISLA(I).LT.0) GO TO 99
                    IP = ISLACK(I)
                    IF(K2.LT.10) WRITE(6,490) ESC, K2, DNOMB(IP), I
490          FORMAT(A, '[', I1, ';10H', 'SE SELECCIONA EL NODO ', A8,
*              ' COMO COMPENSADOR DE LA ISLA ', I2)
                    IF(K2.GE.10) WRITE(6,493) ESC, K2, DNOMB(IP), I
493          FORMAT(A, '[', I2, ';10H', 'SE SELECCIONA EL NODO ', A8,
*              ' COMO COMPENSADOR DE LA ISLA ', I2)
                    K2 = K2 + 2
99 CONTINUE

```

Fin del subprograma ASIGNA.

```

                CALL ENCA
                RETURN
                END

```

f) Subprograma FORMPR

Este subprograma tiene como función el crear la lista encadenada de información acerca de elementos de transmisión, con respecto a los nodos que involucra cada uno de tales elementos.

La forma de efectuar este subprograma es la siguiente:

```
CALL FORMPR (NNODOS)
```

La descripción del argumento involucrado y de las variables del bloque común se muestran a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.
INICPR(NNODOS)	S	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.
IFINPR(2*NLINES)	S	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NE(i) indica el número del nodo de envío.
NR(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	S	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i.

El código de este subprograma es el siguiente:

Inicio del subprograma FORMPR.

```
SUBROUTINE FORMPR (NNODOS)
```

Declaración de bloques comunes y de variables.

```
COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
```

C

```
INTEGER NE(1000),NR(1000),INICPR(550),IFINPR(2000),IPOSPR(2000)
```

C

K es un contador de posiciones de los circuitos π ya incluidos en la lista de localidades contiguas IPOSPR(.), en donde se indica donde se encuentra el circuito π asociado al nodo I.

```
K = 1
```

```
DO 50 I = 1, NNODOS
```

```
    INICPR(I) = K
```

```
    DO 40 J = 1, NLINES
```

```
        IF(I.NE.NE(J).AND.I.NE.NR(J)) GO TO 40
```

Se guarda la posición de un circuito π asociado al nodo I.

```
IPOSPR(K) = J
```

Se incrementa K en 1, indicando que se ha encontrado otro circuito.

```
K = K + 1
```

Se agrega otro circuito asociado al nodo I a la lista encadenada IFINPR

```
IFINPR(K-1) = K
```

```
40 CONTINUE
```

Se termina la información encadenada al nodo I.

```
IFINPR(K-1) = 0
```

```
50 CONTINUE
```

Fin del subprograma FORMPR.

```
RETURN
```

```
END
```

g) Subprograma INICIA

Este subprograma tiene las funciones siguientes:

(a) Especificar condiciones iniciales de magnitud y ángulo de fase nodales.

(b) Calcular las potencias especificadas en cada nodo.

La forma de invocarlo es:

```
CALL INICIA (VRE, VIM, VOLT, ANGULO, PG, PC, QG, QC, NNODOS, QGMIN,
*          QGMAX, NNGEN, VESP, NPV, NISLAS, ISLACK, TIPO, DNOMB,
*          NAU, TIPGEN, INIC, SBASE, INISLA, ISLA)
```

Las variables involucradas en su argumento se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
VRE(NNODOS)	S	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	S	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
VOLT(NNODOS)	S	REAL	Magnitud del voltaje complejo nodal.
ANGULO(NNODOS)	S	REAL	Angulo de fase del voltaje complejo nodal.
PG(NNODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema.
PC(NNODOS)	E	REAL	Potencia real de carga en cada nodo del sistema.
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.
QC(NNODOS)	E	REAL	Potencia reactiva de carga en cada nodo del sistema.
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
QGMIN(NNGEN)	E/S	REAL	Límite inferior de potencia reactiva en cada nodo de generación del sistema.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
QGMAX(NNGEN)	E/S	REAL	Límite superior de potencia reactiva en cada nodo de generación del sistema.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema.
VESP(NNGEN)	E	REAL	Magnitud de voltaje especificada en cada nodo de generación del sistema.
NPV(NNGEN)	E	ENTERA	Contiene el número asignado a cada nodo de generación.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
ISLACK(NISLAS)	S	ENTERA	ISLACK(i) contiene al nodo compensador de la isla i.

El código de este subprograma se muestra a continuación.

Inicio del subprograma INICIA.

```
SUBROUTINE INICIA (VRE, VIM, VOLT, ANGULO, PG, PC, QG, QC, NNODOS, QGMIN,
*                QGMAX, NNGEN, VESP, NPV, NISLAS, ISLACK, INISLA, ISLA)
```

Declaración de variables.

```
DIMENSION VRE(550), VIM(550), PG(550), VESP(150), NPV(150), PC(550),
*          QG(550), QC(550), QGMIN(150), QGMAX(150), ISLACK(10),
*          VOLT(550), ANGULO(550), INISLA(10), ISLA(550)
```

Se asigna a VMAG y ANGU la magnitud y el ángulo de fase valores de perfil plano como voltaje inicial (ANGU en grados).

```
VMAG = 1.0
ANGU = 0.00000001
```

Se convierte ANGU a radianes.

```
ANGU = ANGU * 3.1415927 / 180.0
```

C

Se calcula parte real e imaginaria del voltaje complejo.

```
VR = VMAG * COS(ANGU)
```

```
VI = VMAG * SIN(ANGU)
```

En cada nodo se llenan variables del voltaje complejo con VMAG, ANGU, VR y VI.

```
DO 10 I = 1, NNODOS
    VRE(I) = VR
    VIM(I) = VI
    VOLT(I) = VMAG
    ANGULO(I) = ANGU
10 CONTINUE
```

Se calcula potencias especificadas en cada nodo.

```
DO 20 I = 1, NNODOS
    PG(I) = PG(I) - PC(I)
    QG(I) = QG(I) - QC(I)
20 CONTINUE
```

Se resta a los límites de potencia reactiva generada la potencia de carga en el mismo nodo para agilizar su revisión durante el proceso iterativo.

```
IF(NNGEN.EQ.0) GO TO 40
DO 30 I = 1, NNGEN
    J = NPV(I)
```

Se verifica que el nodo J esté dado de alta.

```
IF (J.LT.0) GO TO 30
```

El nodo J está dado de alta.

```
QGMIN(I) = QGMIN(I) - QC(J)
QGMAX(I) = QGMAX(I) - QC(J)
```

El perfil plano para nodos de generación se afecta por la magnitud de voltaje especificada.

```
VRE(J) = VESP(I) * VR
VIM(J) = VESP(I) * VI
VOLT(J) = VESP(I)
30 CONTINUE
40 DO 50 I = 1, NISLAS
```

Se verifica que la isla actual esté activa.

```
IF (INISLA(I).LT.0) GO TO 47
```

La isla actual está activa (tiene generación de potencia real).
Se asigna el ángulo de 0 grados al nodo compensador de la misma.

```
J = ISLACK(I)
VIM(J) = 0.0
ANGULO(J) = 0.0
```

Se busca el nodo compensador en la lista de nodos de generación para asignar la magnitud de voltaje a la parte real del voltaje.

```
DO 45 II = 1, NNGEN
    JJ = NPV(II)
    IF (JJ.LT.0.OR.JJ.NE.J) GO TO 45
    VRE(J) = VESP(II)
    VOLT(J) = VESP(II)
45    CONTINUE
    GO TO 50
```

Se tiene una isla inactiva. Entonces, el voltaje en todos los nodos de la misma es 0.

```
47    INICIO = ABS(INISLA(I))
    IF (I.LT.NISLAS) IFIN = ABS(INISLA(I + 1)) - 1
    IF (I.EQ.NISLAS) IFIN = NNODOS
    DO 48 JJ = INICIO, IFIN
        NODO = ISLA(JJ)
        VRE(NODO) = 0.0
        VIM(NODO) = 0.0
48    CONTINUE
50    CONTINUE
```

C

Fin del subprograma INICIA.

```
RETURN
END
```

h) Subprograma MODEBP

Este subprograma tiene como función obtener los modelos [B'] y [B''] del método desacoplado rápido. Estos modelos se calculan en forma generalizada, sin considerar tipos de nodos o elementos de transmisión que los modifiquen. Estas variaciones se efectúan posteriormente en el subprograma DERPAR.

La instrucción para ejecutar este subprograma es la siguiente:

```
CALL MODEBP (X, BPRIMA, BBPRIM, NNODOS, INIC, ICOL, ISP, YBUSI, LONG)
```

La descripción de argumentos y bloques comunes asociados se muestra a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
X(NLINES)	E	REAL	Reactancia inductiva serie
BPRIMA(LONG)	S	REAL	Matriz de coeficientes del modelo de potencia activa del método desacoplado rápido.
BBPRIM(LONG)	S	REAL	Matriz de coeficientes del modelo de potencia reactiva del método desacoplado rápido.
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
INIC(NNODOS)	E	ENTERA	Indica la posición en la que inicia la información de cada nodo de la matriz de admitancias nodal en un segundo esquema de empaquetamiento.
LONG	E	ENTERA	$LONG = NNODOS + 2 * NLINES.$
ICOL(LONG)	E	ENTERA	ICOL(k) indica la columna donde se encuentra un elemento de la matriz de admitancias nodal distinto de cero, asociado al renglón i, donde ICOL(k) se puede acceder barriendo la información encadenada a partir de INIC(i).
YBUSI(LONG)	E	REAL	Parte imaginaria de un elemento de la matriz de admitancias nodal.
ISP(LONG)	E	ENTERA	Indica la siguiente posición en la información empaquetada donde se encuentra otro elemento del mismo renglón.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NE(i) indica el número del nodo de envío.
NR(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	E	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i.

El código de este subprograma se muestra a continuación.

Inicio del subprograma MODEBP.

```
C
C   SUBROUTINE PARA FORMAR MODELOS DEL METODO DESACOPLADO RAPIDO.
C
C   SUBROUTINE   MODEBP   (X, BPRIMA, BBPRIM, NNODOS, INIC, ICOL, ISP, YBUSI,
*                   LONG)
```

Declaración de bloques comunes y de variables.

```
C
COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
C
DIMENSION X(1000), NE(1000), NR(1000), INICPR(550), IPOSPR(2000),
*          BBPRIM(2550), YBUSI(2550), ICOL(2550), ISP(2550),
*          BPRIMA(2550), INIC(550), IFINPR(2000)
```

Se inicializan los arreglos BPRIMA y BBPRIM en cero.

```
DO 2 I = 1, LONG
    BPRIMA(I) = 0.0
    BBPRIM(I) = 0.0
```

2 CONTINUE

Se construyen los modelos.

```

DO 30 I = 1, NNODOS
      SUM1 = 0.0
      SUM2 = 0.0
      IN = INIC (I)
      IPOS = IN
10      IPOS = ISP(IPOS)
      IF (IPOS.EQ.0) GO TO 25
      IC = ICOL(IPOS)

```

BBPRIM(.) es el negativo de la parte imaginaria de la matriz de admitancias nodal.

$$\text{SUM2} = \text{SUM2} + \text{YBUSI}(\text{IPOS})$$

Se calcula un elemento fuera de la diagonal de BBPRIM(.).

$$\text{BBPRIM}(\text{IPOS}) = - \text{YBUSI}(\text{IPOS})$$

Se empieza a recorrer la información de elementos de transmisión asociados con el nodo I, debido a que BPRIMA(.) está formada por una matriz de admitancias considerando únicamente la reactancia inductiva serie de cada elemento de transmisión.

```

      JC = INICPR(I)
15      J = IPOSJR(JC)
      IF(IC.NE.NE(J).AND.IC.NE.NR(J)) GO TO 20
      BPQ = - 1.0 / X(J)
      SUM1 = SUM1 - BPQ

```

Se calcula un elemento fuera de la diagonal de BPRIMA(.).

```

      BPRIMA (IPOS) = BPRIMA (IPOS) + BPQ
20      JC = IFINPR(JC)

```

Si se cumple esta condición significa que ya no existen más elementos de transmisión asociados al nodo I.

$$\text{IF} (\text{JC.EQ.0}) \text{ GO TO } 10$$

Existe otro elemento de transmisión asociado al nodo I.

$$\text{GO TO } 15$$

Se calcula un elemento diagonal de BBPRIM(.) y BPRIMA(.).

```

25      BPRIMA (IN) = SUM1
      BBPRIM (IN) = SUM2
30 CONTINUE

```

Fin del subprograma MODEBP.

```

RETURN
END

```


i) Subprograma REINIC

Este subprograma tiene como propósito inicializar nuevamente las variables de control después de haber efectuado un estudio de flujos y entrado al subprograma TOPCAM a realizar algunos cambios.

La forma de ejecutar este subprograma es:

```
CALL REINIC (NNODOS, TIPO, PG, QG, NPV, QGMIN, QGMAX, NNGEN,
*          NSHNTS, NSHUNT, SHUNT, PC, QC, NISLAS, ISLA,
*          INISLA, FPCISLA, FQCISLA, NAU, PSLACK)
```

La descripción de variables del argumento y bloques comunes se muestra a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
TIPO(NNODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
PG(NNODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema.
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema eléctrico.
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.
QGMIN(NNGEN)	E/S	REAL	Límite inferior de potencia reactiva en cada nodo de generación del sistema.
QGMAX(NNGEN)	E/S	REAL	Límite superior de potencia reactiva en cada nodo de generación del sistema.
NSHNTS	E/S	ENTERA	Número total de elementos de compensación fija en derivación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NSHUNT(NSHNTS)	E/S	ENTERA	Identificador de cada nodo en el cual existe un compensador en derivación.
SHUNT(NSHNTS)	E/S	REAL	Susceptancia capacitiva o inductiva en por unidad del compensador en derivación.
PC(NNODOS)	E/S	REAL	Potencia real de carga en cada nodo del sistema.
QC(NNODOS)	E/S	REAL	Potencia reactiva de carga en cada nodo del sistema.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
FPCISLA(NISLAS)	E	REAL	Factor de modificación de la potencia real de carga de los nodos pertenecientes a una isla.
FQCISLA(NISLAS)	E	REAL	Factor para modificar la potencia reactiva de carga de los nodos pertenecientes a una isla.
NAU(NNGEN)	E/S	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
PSLACK(NISLAS)	S	ENTERA	Potencia real nominal de generación del nodo compensador de cada isla.
NLINES	E	ENTERA	Número total de elementos de transmisión en el sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
INICPR(NNODOS)	E	ENTERA	INICPR(i) indica el lugar donde inicia la información referente a la posición de circuitos π , la cual se localiza en IPOSPR(.), correspondiente al nodo i.
IFINPR(2*NLINES)	E	ENTERA	Encadena la información de circuitos π relacionados con el nodo i, iniciando en la posición marcada por INICPR(i).
NE(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NE(i) indica el número del nodo de envío.
NR(NLINES)	E	ENTERA	Para el circuito π i-ésimo, NR(i) indica el número del nodo de recepción.
IPOSPR(2*NLINES)	E/S	ENTERA	IPOSPR(k), con k definida en IFINPR(.), indica donde se encuentra un circuito π relacionado con el nodo i.

El código de este subprograma se muestra a continuación.

Inicio del subprograma REINIC.

```

C   SUBROUTINA DE REINICIALIZACION.
C
C   SUBROUTINE REINIC (NNODOS, TIPO, PG, QG, NPV, QGMIN, QGMAX, NNGEN,
*                   NSHNTS, NSHUNT, SHUNT, PC, QC, NISLAS,
*                   ISLA, INISLA, FPCISLA, FQCISLA, NAU, PSLACK)
C

```

Declaración de bloques comunes y de variables.

```

C   COMMON /CIRCU1/ NLINES, INICPR, IFINPR, NE, NR, IPOSPR
C
C   INTEGER NE(1000), NR(1000), NSHUNT(150), NPV(150), TIPO(550),
*          NAU(150), INISLA(10), INICPR(550), IPOSPR(2000),
*          ISLA(550), IFINPR(2000)
C
C   REAL PG(550), QGMAX(150), PSLACK(10), QG(550), QGMIN(150),
*        SHUNT(150), PC(550), QC(550), FPCISLA(10), FQCISLA(10)
C

```

Se reinicializa el arreglo IPOSPR(.).

```
DO 20 I = 1, NLINES * 2
      IPOSPR(I) = ABS (IPOSPR(I) )
20 CONTINUE
```

Se reinicializa la variable del tipo de nodo en generadores (TIPO(.)=2)

```
DO 25 I = 1, NNGEN
      J = NPV(I)
```

Este generador fué dado de baja al cumplirse esta condición.

```
IF (J.LT.0) GO TO 25
```

Este generador tuvo algún límite violado de potencia reactiva generada.

```
IF (TIPO(J).EQ.3) TIPO(J) = 2
25 CONTINUE
```

Se verifica que algún compensador en derivación exista por violación de límites en un compensador estático de potencia reactiva. Si es así, se restablece el tipo de nodo y se elimina tal compensador de los arreglos que contienen esta información.

```
L = 1
DO 30 I = 1, NSHNTS
      J = NSHUNT(I)
      IND = 0
```

Si $J < 0$, el compensador ha sido dado de baja.

```
IF (J.LT.0) GO TO 28
```

Se busca que no sea el nodo del compensador un CEV con límites violados. Si ésto ocurre, IND sale con valor de 1.

```
DO 26 K = 1, NNGEN
      IF (NAU(K).GT.0) GO TO 26
      IF (ABS(NAU(K)).EQ.J) IND = 1
26 CONTINUE
```

Se elimina el compensador de la lista si es necesario.

```
28 NSHUNT(L) = NSHUNT(I)
   SHUNT(L) = SHUNT(I)
   IF (IND.EQ.0) L = L + 1
30 CONTINUE
```

Se actualiza el valor de NSHNTS.

```
NSHNTS = L - 1
```

En este ciclo iterativo, se actualizan cargas en cada isla eléctrica de acuerdo a los valores dados en FPCISLA(.) y FQCISLA(.).

```
DO 90 I = 1, NISLAS
```

```

K = ABS(INISLA(I))
FACTP = FPCISLA(I)
FACTQ = FQCISLA(I)
IF(I.LT.NISLAS) L = ABS(INISLA(I + 1)) - 1
IF(I.EQ.NISLAS) L = NNODOS
DO 85 J = K, L
    J1 = ABS(ISLA(J))
    PC(J1) = PC(J1) * FACTP
    QC(J1) = QC(J1) * FACTQ

```

Se reinician potencias en los nodos que se seleccionaron como compensadores en el estudio de flujos anterior. También, se reasigna el tipo de nodo generador (TIPO(.) = 2).

```

IF(TIPO(J1).EQ.1) PG(J1) = PSLACK(I)
IF(TIPO(J1).EQ.1) TIPO(J1) = 2
IF(TIPO(J1).EQ.2) QG(J1) = 0.0
ISLA(J) = J1
85 CONTINUE
90 CONTINUE

```

Fin del subprograma REINIC.

```

RETURN
END

```

j) Subprograma SOLFLU

La función de este subprograma es efectuar el proceso iterativo del método desacoplado rápido, el cual está representado por las siguientes actividades:

- Ordenamiento del sistema de ecuaciones lineales.
- Cálculo de desviaciones de potencia.
- Construcción y factorización de las matrices BPRIMA y BBPRIM.
- Resolución del sistema de ecuaciones lineales en cada iteración.
- Revisión de límites de potencia reactiva generada.
- Actualización de las variables de estado (voltajes y ángulos).

Además, informa acerca de la evolución del proceso iterativo, hasta lograr la convergencia hacia una solución o, en el peor de los casos, la divergencia.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL SOLFLU (NNODOS, TIPO, TOLP, TOLQ, PG, QG, NMAXIT, IERR, NEXT, VIM,
*          VOLT, ANGULO, INIC, ICOL, YBUSR, YBUSI, ISP, PCAL, QCAL,
*          NPV, NNGEN, DNOMB, QGMIN, QGMAX, NCEVS, PC, QC, NORDEN,
*          JINIC, JICOL, JISP, IFACT, VRE, JL, VESP, KISLA, NISLAS,
*          INISLA, ISLA, NAU, ISLACK, YM, NLT, LONG, TOLREV, BPRIMA,
*          BBPRIM, NSHNTS, NSHUNT, SHUNT, KFUERA, NODFUE)
```

cuyos argumentos y bloques comunes se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
TIPO(NNODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
TOLP	E	REAL	Tolerancia máxima de error en desviaciones de inyecciones nodales de potencia real (activa).
TOLQ	E	REAL	Tolerancia máxima de error en desviaciones de inyecciones nodales de potencia reactiva.
PG(NNODOS)	E/S	REAL	Potencia real de generación en cada nodo del sistema.
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.
NMAXIT	E	ENTERA	Número de iteraciones máximo del proceso iterativo.
IERR	E/S	ENTERA	IERR = 1 si el proceso iterativo no converge a alguna solución.
PC(NNODOS)	E	REAL	Potencia real de carga en cada nodo del sistema.
QC(NNODOS)	E	REAL	Potencia reactiva de carga en cada nodo del sistema.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NEXT(JL)	E/S	ENTERA	Arreglo para encadenar posiciones de elementos distintos de cero en un renglón de la matriz BPRIMA o de la BBPRIM, después de haber efectuado el ordenamiento del sistema de ecuaciones.
VRE(NNODOS)	E/S	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	E/S	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
VOLT(NNODOS)	E/S	REAL	Magnitud del voltaje complejo en cada nodo del sistema eléctrico.
ANGULO(NNODOS)	E/S	REAL	Angulo de fase en grados del voltaje complejo en cada nodo del sistema eléctrico.
INIC(NNODOS)	E	ENTERA	Indica la posición en que inicia la información de cada renglón de la matriz de admitancias nodal.
ICOL(LONG)	E	ENTERA	Contiene información de la columna asociada con un renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal distinto de cero.
LONG	E	ENTERA	$LONG = NNODOS + 2 * NLINES$
YBUSR(LONG)	E	ENTERA	Parte real de elementos de la matriz de admitancias nodal.
YBUSI(LONG)	E/S	ENTERA	Parte imaginaria de los elementos de la matriz de admitancias nodal.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ISP(LONG)	E	ENTERA	Lista para encadenar la información relevante de cada renglón de la matriz de admitancias nodal.
PCAL(NNODOS)	E/S	REAL	Potencia real neta inyectada en cada nodo del sistema eléctrico.
QCAL(NNODOS)	E/S	REAL	Potencia reactiva neta inyectada en cada nodo del sistema eléctrico.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema eléctrico.
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.
QGMIN(NNGEN)	E/S	REAL	Límite inferior de potencia reactiva en cada nodo de generación del sistema.
QGMAX(NNGEN)	E/S	REAL	Límite superior de potencia reactiva en cada nodo de generación del sistema.
DNOMB(NNODOS)	E	CHARACTER	Contiene el nombre de cada nodo del sistema eléctrico.
NCEVS	E	ENTERA	Número total de compensadores estáticos de potencia reactiva.
NORDEN(NNODOS)	E/S	ENTERA	Contiene el orden por renglones para efectuar la factorización de BPRIMA y BBPRIM.
JINIC(NNODOS)	E/S	ENTERA	Indica la posición donde se inicia la información de cada renglón de las matrices BPRIMA y BBPRIM.
JICOL(LONG)	E/S	ENTERA	Indica la columna asociada al renglón definido por JINIC(.), y que en conjunto marcan la localización de un elemento distinto de cero en BPRIMA y BBPRIM.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
JISP(LONG)	E/S	ENTERA	Contiene listas encadenadas de información relevante en cada renglón de BPRIMA y BBPRIM.
IFACT(NNODOS)	E/S	ENTERA	Contiene la posición en que inicia la información relevante de cada renglón de BPRIMA y BBPRIM, después de haber efectuado el ordenamiento.
JL	S	ENTERA	JL=LONG + elementos creados durante la factorización.
VESP(NNGEN)	E	REAL	Voltaje especificado en las terminales de cada nodo de generación.
KISLA	E	ENTERA	Identifica a la isla que se le está haciendo el estudio de flujos.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
NAU(NNGEN)	E/S	ENTERA	Nodo de alta tensión asociado al nodo generador indicado en la misma posición de NPV(.).
ISLACK(ISLAS)	E	ENTERA	En cada posición contiene el nodo compensador de una isla eléctrica.
YM(NNGEN)	S	REAL	Susceptancia capacitiva o inductiva equivalente a los límites violados de un compensador estático de potencia reactiva.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NLT(NNODOS)	E/S	ENTERA	Indica si el nodo asociado a cada posición contiene compensación fija en derivación. En este caso, $NLT(i) = 1$, donde i es el nodo asociado.
LONG	E	ENTERA	$LONG = NNODOS + 2 * NLINES$
TOLREV	E	REAL	Tolerancia máxima de error en desviaciones de potencia reactiva para iniciar la revisión de límites en generadores.
BPRIMA(LONG)	E/S	REAL	Matriz de coeficientes del modelo de potencia activa del método desacoplado rápido.
BBPRIM(LONG)	E/S	REAL	Matriz de coeficientes del modelo de potencia reactiva del método desacoplado rápido.
NSHNTS	E/S	ENTERA	Número total de elementos de compensación fija en derivación.
NSHUNT(NSHNTS)	E/S	ENTERA	Identificador de cada nodo en el cual existe un compensador en derivación.
SHUNT(NSHNTS)	E/S	REAL	Susceptancia capacitiva o inductiva en por unidad del compensador en derivación.
KFUERA	E	ENTERA	Contador de nodos dados de baja en el subprograma TOPCAM.
NODFUE(KFUERA)	E	ENTERA	Número de cada nodo que ha sido dado de baja en TOPCAM.
NCEVV(NCEVS)	S	ENTERA	Indica el nodo de voltaje controlado tipo CEV, cuyos límites de potencia reactiva han sido violados.

El código de este subprograma es mostrado a continuación.

Inicio del subprograma SOLFLU.

```

SUBROUTINE SOLFLU (NNODOS, TIPO, TOLP, TOLQ, PG, QG, NMAXIT, IERR, NEXT,
*                VIM, VOLT, ANGULO, INIC, ICOL, YBUSR, YBUSI, ISP, PCAL,
*                QCAL, NPV, NNGEN, DNOMB, QGMIN, QGMAX, NCEVS, PC, QC,
*                NORDEN, JINIC, JICOL, JISP, IFACT, VRE, JL, VESP,
*                KISLA, NISLAS, INISLA, ISLA, NAU, ISLACK, YM, NLT,
*                LONG, TOLREV, BPRIMA, BBPRIM, NSHNTS, NSHUNT, SHUNT,
*                KFUERA, NODFUE)

```

C

Declaración de bloques comunes y de variables.

```

COMMON /CIRCU1/ NLines, INICPR, IFINPR, NE, NR, IPOSPR

```

C

```

DIMENSION INIC(550), ICOL(2550), YBUSR(2550), YBUSI(2550), PC(550),
*         ISP(2550), QC(550), PG(550), QG(550), PCAL(550), YM(150),
*         QCAL(550), VOLT(550), ANGULO(550), DELTAP(550), NE(1000),
*         DELTAQ(550), JICOL(2550), VESP(150), NPV(150), NLT(550),
*         JISP(2550), QGMIN(150), QGMAX(150), IFACT(550), NR(1000),
*         NORDEN(550), VRE(550), VIM(550), NEXT(2550), INISLA(10),
*         NCEVV(150), ISLA(550), ISLACK(10), NAU(150), JINIC(550),
*         BPRIMA(2550), BBPRIM(2550), INPV(150), INICPR(550),
*         NSHUNT(150), SHUNT(150), IFINPR(2000), IPOSPR(2000),
*         NODFUE(15)

```

C

```

INTEGER TIPO(550)

```

C

```

REAL JAH(2550), JAL(2550)

```

C

```

CHARACTER*8 DNOMB(550)
CHARACTER ESC*1
CHARACTER WHITEFORE*4, YELFORE*4
PARAMETER (YELFORE='[33m', WHITEFORE = '[37m')

```

C

```

ESC = CHAR(27)

```

Se inicializa IERR tal que el proceso de solución termina convergiendo.

```

IERR = 0

```

Se inicializan potencias netas inyectadas en cada nodo.

```

DO I = 1, NNODOS
  PCAL(I) = 0.0
  QCAL(I) = 0.0
END DO

```

Se inicializa el vector INPV(.) en 0, indicando que no existen límites de potencia reactiva de generación violados.

```

DO 500 I = 1, NNGEN
500 INPV(I) = 0

```

Se inicializa el vector NCEVV(.) indicando que no existen violaciones de límites de potencia reactiva en estos elementos.

```

DO 507 I = 1, NCEVS
507   NCEVV(I) = 0
      CALL CLEAR
      IF (NISLAS.EQ.1) GO TO 35

```

Se imprime en pantalla letrero de notificación del proceso de solución en la isla actual. Este letrero se imprime cuando NISLAS > 1.

```

25 WRITE (6,30) KISLA
30 FORMAT (15X,' PROCESO ITERATIVO DE LA ISLA ',I2,/)

```

Se imprime encabezado para estar notificando la evolución del proceso iterativo.

```

35 WRITE(6,47)
47 FORMAT(3X,' ITERACION',8X,'NODO',5X,'Max. Desv. en P',7X,'NODO',
*          5X,'Max. Desv. en Q',/)

```

IND es un indicador de la opción de verificación de límites de potencia reactiva. Si IND = 0, se revisan límites. Si IND = 1, no se revisan. Esto, se define por los valores de TOLQ y TOLREV.

```

IND = 0
IF (TOLQ.GT.TOLREV) IND = 1

```

Se inicializa contador de iteraciones e indicador de violación de límites de potencia reactiva.

```

KITER = 0
KVC = 0

```

Se entra al subprograma de ordenamiento de las matrices BPRIMA y BBPRIM de acuerdo al segundo esquema.

```

CALL ORDENA (NNODOS, INIC, ICOL, ISP, JINIC, JICOL,
*           JISP, NORDEN, IFACT, NEXT, JL, LONG)

```

C

Se modifican BPRIMA y BBPRIM para modelar únicamente la isla actual. Tales modificaciones se almacenan en JAH y JAL, respectivamente.

```

CALL DERPAR (NNODOS, INIC, ICOL, BPRIMA, BBPRIM, ISP, JAH, JAL, JL,
*           TIPO, NISLAS, INISLA, ISLA, KISLA, NNGEN, NPV, NAU)

```

C

Factorización LU del modelo de potencia activa, JAH(.).

```

CALL FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAH, JISP, IFACT, NEXT, JL)

```

C

Factorización LU del modelo de potencia reactiva, JAL(.).

```

CALL FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAL, JISP, IFACT, NEXT, JL)

```

C

Cálculo de potencias netas inyectadas en cada nodo.

```
50 CALL PNODAL (NNODOS, INIC, ICOL, YBUSR, YBUSI, ISP, VRE, VIM,
* PCAL, QCAL, NISLAS, INISLA, ISLA, KISLA, ISLACK)
```

C

Cálculo de desviaciones de potencias nodales.

```
CALL DESVIA (PCAL, QCAL, PG, QG, DMAXP, DMAXQ, NK1, NK2, DELTAP,
* DELTAQ, NNODOS, TIPO, VESP, VRE, VIM, NPV, NNGEN, NISLAS,
* INISLA, ISLA, KISLA, ISLACK, NAU, KFUERA, NODFUE)
```

Se guarda la máxima desviación de potencia activa y el nodo donde se presentan, en las variables DMAXPB y NKP.

```
DMAXPB = DMAXP
NKP = NK1
```

Se efectúa una iteración del problema de potencia activa.

```
75 CALL SOLBBP (NNODOS, NORDEN, JICOL, JAH, DELTAP, IFACT, NEXT, JL)
```

C

KA = 1 indica que en CORRIG solamente se actualizan valores de ángulo.

```
KA = 1
```

Actualización de variables de estado.

```
CALL CORRIG (NNODOS, VRE, VIM, VOLT, ANGULO, DELTAP, DELTAQ, KA)
```

Se calculan nuevamente potencias netas inyectadas nodales.

```
CALL PNODAL (NNODOS, INIC, ICOL, YBUSR, YBUSI, ISP, VRE, VIM,
* PCAL, QCAL, NISLAS, INISLA, ISLA, KISLA, ISLACK)
```

Se calculan desviaciones de potencia nodal.

```
CALL DESVIA (PCAL, QCAL, PG, QG, DMAXP, DMAXQ, NK1, NK2, DELTAP,
* DELTAQ, NNODOS, TIPO, VESP, VRE, VIM, NPV, NNGEN, NISLAS,
* INISLA, ISLA, KISLA, ISLACK, NAU, KFUERA, NODFUE)
```

Se guarda la máxima desviación de potencia reactiva y el nodo donde se presentan, en las variables DMAXQB y NKQ.

```
DMAXQB = DMAXQ
NKQ = NK2
```

La primera condición verifica si el usuario optó por revisar límites de potencia reactiva. La segunda condición revisa que existan nodos PV. La tercer condición revisa que el proceso iterativo se haya acercado lo suficiente, según el usuario, a la solución para iniciar la revisión de límites.

```
IF (IND.EQ.1) GO TO 54
IF (NNGEN.EQ.0) GO TO 54
IF (DMAXQ.GT.TOLREV) GO TO 54
```

C
 C SE REVISAN LIMITES DE GENERACION DE POTENCIA REACTIVA.
 C

KVC sale con valor distinto de 0 si existe violación de límites.

KVC = 0

C
 CALL REVNPV (NNODOS, TIPO, NNGEN, NPV, QGMIN, QGMAX, VESP, INPV, QG, QC,
 * QCAL, DELTAQ, VOLT, KVC, NISLAS, INISLA, ISLA, KISLA, NAU,
 * NCEVS, NCEVV, YM, INIC, ICOL, YBUSI, ISP, DELTAP, NLT,
 * ISLACK)

C

Se verifica el valor de KVC.

IF (KVC.EQ.0) GO TO 54

C

Se han violado límites de pot. reactiva, por lo que se reconstruyen los modelos del método desacoplado rápido.

CALL DERPAR (NNODOS, INIC, ICOL, BPRIMA, BBPRIM, ISP, JAH, JAL, JL,
 * TIPO, NISLAS, INISLA, ISLA, KISLA, NNGEN, NPV, NAU)

C

Se refactorizan los modelos.

CALL FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAH, JISP, IFACT, NEXT, JL)
 CALL FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAL, JISP, IFACT, NEXT, JL)

Se imprimen las desviaciones de potencia en la iteración actual.

54 WRITE(6,56) KITER, DNOMB(NKP), DMAXPB, DNOMB(NKQ), DMAXQB
 56 FORMAT(6X, I2, 10X, A8, 4X, F13.6, 6X, A8, 2X, F13.6)

Esta condición se refiere a la posibilidad de haber iniciado apenas el proceso iterativo y revisado límites de potencia reactiva con resultado de alguna violación.

IF (KITER.LE.3.OR.KVC.EQ.1) GO TO 63

Esta condición se refiere a la posibilidad de divergencia.

IF (DMAXPB.GT.1000.0.OR.DMAXQB.GT.1000.0)GO TO 80

Se verifican tolerancias especificadas y resultado de la revisión de límites de potencia reactiva generada.

63 IF (DMAXPB.GT.TOLP.OR.DMAXQB.GT.TOLQ.OR.KVC.NE.0) GO TO 60

Se han pasado las pruebas de convergencia.

WRITE (6,55) KITER
 55 FORMAT (//10X, 'EL ESTUDIO CONVERGE EN ', I3, ' ITERACIONES', //)
 CALL ENCA
 GO TO 100

Se verifica que el máximo de iteraciones no se haya rebasado.

```
60 IF (KITER.GE.NMAXIT) GO TO 80
```

C

Se resuelve el problema de potencia reactiva.

```
CALL SOLBBP (NNODOS, NORDEN, JICOL, JAL, DELTAQ, IFACT, NEXT, JL)
```

C

Actualización de variables de estado.

```
KA = 0
```

```
CALL CORRIG (NNODOS, VRE, VIM, VOLT, ANGULO, DELTAP, DELTAQ, KA)
```

C

Se incrementa el contador de iteraciones en 1 y se regresa a efectuar otra iteración.

```
KITER = KITER + 1
```

```
GO TO 50
```

C

El proceso de solución no converge en el número de iteraciones especificado.

```
80 WRITE(6,90) KITER
```

```
90 FORMAT(//10X, 'EL ESTUDIO NO CONVERGE EN ', I2, ' ITERACIONES')
```

```
CALL ENCA
```

```
IERR = 1
```

```
100 CONTINUE
```

Al terminar el proceso iterativo se realizan algunos cálculos para facilitar la creación de reportes de resultados.

C

C SE 'CALCULAN' LAS POTENCIAS DE GENERACION DE LA ISLA ACTUAL

C

Se calculan potencias de generación en el nodo compensador y se restablecen en los nodos PV.

```
INICIO = INISLA(KISLA)
```

```
IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
```

```
IF (KISLA.EQ.NISLAS) IFIN = NNODOS
```

```
DO 110 J = INICIO, IFIN
```

```
    I = ABS(ISLA(J))
```

```
    IF (TIPO(I).LT.0) GO TO 110
```

```
    IF (TIPO(I).NE.1) GO TO 107
```

Se calcula potencias de generación en el nodo compensador.

```
PG(I) = PCAL(I) + PC(I)
```

```
QG(I) = QCAL(I) + QC(I)
```

```
GO TO 108
```

```
107 PG(I) = PG(I) + PC(I)
```

Se calcula la potencia reactiva generada en un nodo PV con límites violados.

```

                IF (TIPO(I).EQ.3) QG(I) = QG(I) + QC(I)
                IF (TIPO(I).EQ.2) QG(I) = QCAL(I) + QC(I)
108             TIPO(I) = - TIPO(I)
110 CONTINUE
C
C     SE ORDENAN LOS NODOS DE LA ISLA ACTUAL EN FORMA ASCENDENTE.

```

La forma en que se encadena la información de una isla es por recorrido de elementos de transmisión, quedando en "desorden", con respecto al número que guarda cada nodo al ser leído. Aquí, se ordenan, tal que en los reportes de resultados aparezcan los nodos en un orden semejante al establecido inicialmente.

```

DO I = 1, NNODOS
    TIPO(I) = ABS(TIPO(I))
END DO
    DO 130 I = INICIO, IFIN - 1
        J = ISLA(I)
        IF (J.LT.0) GO TO 130
        IAUX = J
        DO 120 K = I + 1, IFIN
            L = ISLA(K)
            IF (L.LT.0) GO TO 120
            IF(IAUX.LE.L) GO TO 120
            ISLA(K) = IAUX
            IAUX = L
120             CONTINUE
                ISLA(I) = IAUX
130             CONTINUE

```

Se verifica que no haya algún compensador estático de Vars con límites violados. La revisión se efectúa recorriendo todos los nodos de la isla

```

DO 158 JS = INICIO, IFIN
    I = ISLA(JS)

```

Se verifica que el nodo no esté dado de baja.

```

IF (I.LT.0) GO TO 158

```

Se busca al nodo en la lista de nodos de voltaje controlado.

```

DO 157 L = 1, NNGEN
    J = NPV(L)
    IF (J.NE.I) GO TO 157

```

Se ha encontrado al nodo en la lista. Ahora, se restablecen sus límites de potencia reactiva generada.

```

J = TIPO(I)
QGMIN(L) = QGMIN(L) + QC(I)
QGMAX(L) = QGMAX(L) + QC(I)

```


Se revisa si sus límites fueron violados.

IF (J.NE.3) GO TO 158

Sus límites fueron violados y se busca el tipo de nodo de que se trata.

IF(NAU(L).GT.0) GO TO 158

Se trata de un CEV por lo que se incrementa el número de compensadores fijos en derivación y se agrega el nodo de alta tensión y el valor de la susceptancia equivalente al límite violado en las listas correspondientes.

J = ABS(NAU(L))
 NSHNTS = NSHNTS + 1
 NSHUNT(NSHNTS) = J
 SHUNT(NSHNTS) = YM(L)
 INDF = 0

Se da de baja el nodo PV reducido de la isla actual.

ISLA(JS) = - I
 I1 = I

Se dan de baja los circuitos π asociados al nodo PV reducido. INDF indica que se ha dado de baja la lista encadenada del nodo I, al momento de tomar el valor de 1. Toma el valor de 2 cuando se han dado de baja los circuitos de otras listas que contengan a I.

143 JC = INICPR(J)
 144 J1 = IPOSPR(JC)
 IC = NR(J1)
 IF(J.NE.NE(J1)) IC = NE(J1)
 145 IF (IC.EQ.I1) IPOSPR(JC) = - J1
 146 JC = IFINPR(JC)
 IF (JC.NE.0) GO TO 144
 INDF = INDF + 1
 IF (INDF.EQ.2) GO TO 158
 I1 = J
 J = NPV(L)
 GO TO 143
 157 CONTINUE
 158 CONTINUE

Fin del subprograma SOLFLU.

RETURN
 END

k) Subprograma ORDENA

Este subprograma tiene como función ordenar los sistemas de ecuaciones lineales que se resuelven en cada iteración del proceso de solución. Este ordenamiento simula el proceso de eliminación Gaussiana utilizando una gráfica no dirigida de sistema de ecuaciones. El modelo de potencia activa es distinto al de potencia reactiva. Sin embargo, para este simulador se han hecho modificaciones a éste último para usar los mismos apuntadores resultantes de este ordenamiento en ambos modelos.

El esquema de ordenamiento utilizado [2] consiste en lo siguiente:

- (a) Crear una gráfica no dirigida que represente a ambos modelos de potencia.
- (b) Seleccionar al nodo por eliminar a áquel con el menor número de nodos interconectados.
- (c) Simular la eliminación del nodo, desconectándolo de la gráfica e interconectando todos los nodos adyacentes al mismo, creando elementos de interconexión únicamente en el caso de que no existan previamente.
- (d) Regresar al paso (b).

La manera de ejecutar este subprograma es mediante la instrucción:

```
CALL ORDENA (NNODOS, INIC, ICOL, ISP, JINIC, JICOL,
*           JIÍSP, NORDEN, IFACT, NEXT, JL, LONG)
```

A continuación se describen las variables involucradas en su argumento.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
INIC(NNODOS)	E	ENTERA	Indica la posición en que inicia la información de cada renglón de la matriz de admitancias nodal.
LONG	E	ENTERA	LONG = NNODOS + 2*NLINES
JL	S	ENTERA	JL=LONG + elementos creados durante la factorización.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
ICOL(LONG)	E	ENTERA	Contiene información de la columna asociada con un renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal distinto de cero.
ISP(LONG)	E	ENTERA	Lista para encadenar la información relevante de cada renglón de la matriz de admitancias nodal.
NORDEN(NNODOS)	E/S	ENTERA	Contiene el orden por renglones para efectuar la factorización de BPRIMA y BBPRIM.
JINIC(NNODOS)	E/S	ENTERA	Indica la posición donde se inicia la información de cada renglón de las matrices BPRIMA y BBPRIM.
JICOL(LONG)	E/S	ENTERA	Indica la columna asociada al renglón definido por JINIC(.), y que en conjunto marcan la localización de un elemento distinto de cero en BPRIMA y BBPRIM.
JISP(LONG)	E/S	ENTERA	Contiene listas encadenadas de información relevante en cada renglón de BPRIMA y BBPRIM.
IFACT(NNODOS)	E/S	ENTERA	Contiene la posición en que inicia la información relevante de cada renglón de BPRIMA y BBPRIM, después de haber efectuado el ordenamiento.
NEXT(JL)	E/S	ENTERA	Arreglo para encadenar posiciones de elementos distintos de cero en un renglón de la matriz BPRIMA o de la BBPRIM, después de haber efectuado el ordenamiento del sistema de ecuaciones.

El código de este subprograma se muestra a continuación.

Inicio del subprograma ORDENA.

```
C
C   RUTINA DE ORDENAMIENTO DEL SISTEMA DE ECUACIONES LINEALES.
C   S E G U N D O   E S Q U E M A   D E   O R D E N A M I E N T O .
C
C   SUBROUTINE ORDENA (NNODOS, INIC, ICOL, ISP, JINIC, JICOL, JISP,
*                       NORDEN, IFACT, NEXT, JL, LONG)
C
```

Declaración de variables.

```

DIMENSION INIC(550), ICOL(2550), ISP(2550), JINIC(550), NODFUE(15),
*          JICOL(2550), JISP(2550), IFACT(550), NEXT(2550),
*          IAUX(550), NORDEN(550)
C
C   SE GENERAN LOS APUNTADES PARA EL JACOBIANO.
C
```

I1 es una variable auxiliar para almacenar el valor de J, antes de eliminar un nodo (o renglón). J es un contador que indica el número de elementos creados durante el proceso de eliminación.

```
I1 = 1
J = 1
```

Se inicializan los apuntadores de BPRIMA y BBPRIM con los de la matriz de admitancias nodal.

```

DO 5 I = 1, NNODOS
    JINIC(I) = INIC(I)
5 CONTINUE
DO 10 I = 1, LONG
    JICOL(I) = ICOL(I)
    JISP(I) = ISP(I)
10 CONTINUE
C
C   SE ORDENAN LOS RENGLONES DEL JACOBIANO.
C
```

Se inicializa el vector de ordenamiento.

```
DO 20 I = 1, NNODOS
20  NORDEN(I) = 0
```

J se inicia con la posición posterior a la longitud marcada por LONG, definida al formarse la matriz de admitancias nodal.

```
J = LONG + 1
I1 = J
NN1 = NNODOS
```

Se inicia proceso de ordenamiento. Primeramente, se busca al nodo con el menor número de interconexiones. Esto equivale a buscar un renglón en la matriz de coeficientes con el menor número de elementos distintos de cero fuera de la diagonal.

```

DO 60  I11 = 1, NNODOS
      K1 = 1000
DO 30  I = 1, NNODOS
      K = JINIC(I)
      J1 = JICOL(K)
      IF(J1.LT.0.OR.J1.GE.K1) GO TO 30
      K2 = K
      K1 = J1
      I2 = I
30      CONTINUE

```

El nodo seleccionado a eliminar se almacena en I2. Ahora, se buscan interconexiones entre los nodos adyacentes al eliminado.

```

35      K = K2
      K = JISP(K)
      IF(K.EQ.0) GO TO 55
      J2 = JICOL(K)
      L2 = JINIC(J2)
      JI = JICOL(L2)

```

Se resta 1 al nodo interconectado con el que se está eliminando.

```

      IF(JI.LT.0)GO TO 35
      JICOL(L2) = JI - 1
      K3 = K

```

Se busca interconexión entre dos nodos adyacentes al eliminado.

```

40      K3 = JISP(K3)
      IF(K3.EQ.0) GO TO 35
      J3 = JICOL(K3)
      L3 = JINIC(J3)
      JI = JICOL(L3)

```

Se revisa que los nodos que se están visitando no hayan sido eliminados

```

      IF(JI.LT.0) GO TO 40

```

Ahora, se verifica interconexión.

```

45      M2 = L2
      M2 = JISP(M2)
      IF(M2.EQ.0) GO TO 50

```

Si se cumple esta condición ya existe la interconexión.

```

      IF(JICOL(M2).EQ.J3) GO TO 40
      KK = M2
      GO TO 45

```

No hay interconexión, por lo que se crean los apuntadores para el nuevo elemento que interconectará al par de nodos visitados.

```

50      JISP(KK) = J
        JICOL(J) = J3
        JISP(J) = 0
        J = J + 1
        L3 = JINIC(J3)
51      L3 = JISP(L3)
        IF(L3.EQ.0) GO TO 52
        KK = L3
        GO TO 51
52      JISP(KK) = J
        JICOL(J) = J2
        JISP(J) = 0
        J = J + 1
        L2 = JINIC(J2)
        L3 = JINIC(J3)
        JICOL(L2) = JICOL(L2) + 1
        JICOL(L3) = JICOL(L3) + 1
        GO TO 40

```

Se integra el nodo eliminado al vector de ordenamiento. JICOL(.) de la posición diagonal se hace -1 para indicar que ya se eliminó este nodo o renglón.

```

55      NORDEN(II1) = I2
        JI = JINIC(I2)
        JICOL(JI) = -1
60 CONTINUE

```

Se actualiza el valor de la longitud total que deberán tener los arreglos BPRIMA y BBPRIM (y por lo tanto, JICOL y JISP).

```

      JL = J - 1
      IF(J.EQ.I1) JL = J

```

C

C Se crean los arreglos 'IFACT' y 'NEXT', los cuales nos indican el orden
 C para factorizar cada renglón.

C

Los arreglos IFACT y NEXT son semejantes a JINIC y JISP, pero almacenan apuntadores para efectuar la factorización LU de BPRIMA y BBPRIM simulando el intercambio de renglones y columnas de acuerdo al ordenamiento efectuado arriba.

```

      DO 120 I = 1, NNODOS
        DO 65 LI = 1, NNODOS
95      IAUX(LI) = 0
        N1 = NORDEN(I)

```

IAUX(.) se llena con las posiciones distintas de cero de un renglón, las cuales servirán para formar los arreglos IFACT y NEXT.

```

      IAUX(I) = N1
      J = JINIC(N1)
      K = J
70     K = JISP(K)
      IF(K.EQ.0)GO TO 90
      K1 = JICOL(K)
      K2 = 1
75     IF(NORDEN(K2).EQ.K1) GO TO 80
      K2 = K2 + 1
      GO TO 75
80     IAUX(K2) = K1
      GO TO 70

```

Se investiga si N1 está antes o en la diagonal del renglón. IND=0 hasta que se llena IFACT(.) para el renglón actual.

```

90     IND = 0
      DO 110 L = 1, NNODOS
          L1 = IAUX(L)
          IF(L1.EQ.0) GO TO 110
          IF(IND.EQ.0) GO TO 95
          IF(L1.EQ.N1) GO TO 105
          K1 = J
92     K1 = JISP(K1)
          IF(L1.NE.JICOL(K1)) GO TO 92

```

Se almacena la siguiente posición de N1 fuera de la diagonal.

```

          NEXT(K) = K1
          K = K1
          GO TO 110
95     K = J
          IND = 1

```

La primera posición del renglón N1 está antes de la diagonal.

```

          IFACT(I) = J
          IF(L1.EQ.N1) GO TO 110

```

Se busca la primera posición a acceder del nodo actual I.

```

100    K = JISP(K)
          IF(L1.NE.JICOL(K)) GO TO 100

```

La primera posición del renglón N1 está en la diagonal.

```

          IFACT(I) = K
          GO TO 110

```

Se almacena la posición diagonal del renglón N1

```

105    NEXT(K) = J
          K = J
110    CONTINUE

```

Se finaliza la lista encadenada del nodo actual N1.

```

      NEXT(K) = 0
120 CONTINUE

```

Fin del subprograma ORDENA.

```

      RETURN
      END

```

1) Subprograma DERPAR

En el subprograma MODEBP se forman los arreglos BPRIMA(.) y BBPRIM(.) de una manera tal que no se consideran islas en el sistema eléctrico ni el tipo de nodo. En DERPAR, se modifican estos arreglos, almacenándose en JAH(.) y JAL(.), respectivamente, tal que se considera el tipo de nodo y la isla actual para la cual se está resolviendo el estudio de flujos, quedando desconectadas las demás islas, así como el nodo compensador de la isla actual en ambos modelos, así como los nodos PV del modelo de potencia reactiva BBPRIM(.).

La forma de acceder este subprograma es con la instrucción:

```

      CALL DERPAR (NNODOS, INIC, ICOL, BPRIMA, BBPRIM, ISP, JAH, JAL, JL,
*              TIPO, NISLAS, INISLA, ISLA, KISLA, NNGEN, NPV, NAU)

```

cuyos argumentos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
INIC(NNODOS)	E	ENTERA	Indica la posición en que inicia la información de cada renglón de la matriz de admitancias nodal.
ICOL(LONG)	E	ENTERA	Contiene información de la columna asociada con un renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal distinto de cero.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
JL	E	ENTERA	JL=LONG + elementos creados durante la factorización.
ISP(LONG)	E	ENTERA	Lista para encadenar la información relevante de cada renglón de la matriz de admitancias nodal.
BPRIMA(LONG)	E	REAL	Matriz de coeficientes del modelo de potencia activa del método desacoplado rápido.
BBPRIM(LONG)	E	REAL	Matriz de coeficientes del modelo de potencia reactiva del método desacoplado rápido.
JAH(LONG)	E/S	REAL	Matriz de coeficientes del modelo de potencia activa del método desacoplado rápido, para la isla actual del estudio de flujos.
JAL(LONG)	E/S	REAL	Matriz de coeficientes del modelo de potencia reactiva del método desacoplado rápido, para la isla actual del estudio de flujos.
KISLA	E	ENTERA	Identifica a la isla que se le está haciendo el estudio de flujos.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
TIPO(NNODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.
NAU(NNGEN)	E	ENTERA	Nodo de alta tensión asociado al nodo generador indicado en la misma posición de NPV(.).

El código de este subprograma se muestra y describe enseguida.

Inicio del subprograma DERPAR.

```
C
  SUBROUTINE DERPAR (NNODOS, INIC, ICOL, BPRIMA, BBPRIM, ISP, JAH, JAL, JL,
*                   TIPO, NISLAS, INISLA, ISLA, KISLA, NNGEN, NPV, NAU)
C
```

Declaración de variables.

```

  DIMENSION INIC(550), ICOL(2550), BPRIMA(2550), BBPRIM(2550),
*           ISP(2550), INISLA(10), ISLA(550), NPV(150), NAU(150)
C
  INTEGER TIPO(550)
C
  REAL JAH(2550), JAL(2550)
C
```

DIAG es el valor en la diagonal para renglones que no entran en el proceso de solución para la isla actual.

```

  DIAG = 9999999999999999.9999
C
```

Se inicializan JAH(.) y JAL(.), con ceros fuera de la diagonal y con DIAG en ésta.

```

  DO 6 I = 1, JL
    JAH(I) = 0.0
    JAL(I) = 0.0
6 CONTINUE
  DO I = 1, NNODOS
    IPOS = INIC(I)
    JAH(IPOS) = DIAG
    JAL(IPOS) = DIAG
  END DO
```

Se determinan los límites de la lista de localidades contiguas que define a la isla actual.

```

INICIO = INISLA(KISLA)
IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
IF (KISLA.EQ.NISLAS) IFIN = NNODOS

```

Se inicia la construcción de los modelos para la isla actual.

```

DO 50 I = INICIO, IFIN
      J = ISLA(I)

```

Se verifica que el nodo J esté conectado al sistema eléctrico.

```

IF (J.LT.0) GO TO 50
IPOS = INIC(J)
ITIPO = TIPO(J)

```

Se revisa el tipo de nodo que es J. Al cumplirse la siguiente condición el nodo es el compensador de la isla actual y el renglón queda como fue inicializado.

```

IF (ITIPO.EQ.1) GO TO 50

```

Al cumplirse la siguiente condición, el nodo es de voltaje controlado y sólo se modifica el renglón del modelo de potencia activa BPRIMA(.).

```

IF (ITIPO.EQ.2) GO TO 10

```

El nodo J es de carga (tipo 3). Ahora, se verifica que J no sea un CEV con los límites violados (condición dentro del siguiente ciclo).

```

DO 8 L = 1, NNGEN
      IF (J.EQ.NPV(L).AND.NAU(L).LT.0) GO TO 50
8      CONTINUE

```

Se modifica el renglón de cada modelo, de acuerdo al tipo, iniciando por el elemento diagonal (sin considerar ordenamiento).

```

10      JAH(IPOS) = BPRIMA(IPOS)
        IF(ITIPO.EQ.3) JAL(IPOS) = BBPRIM(IPOS)
15      IPOS = ISP(IPOS)
        IF(IPOS.EQ.0) GO TO 50
        J1 = ICOL(IPOS)
        IF (TIPO(J1).EQ.1) GO TO 15
        JAH(IPOS) = BPRIMA(IPOS)
        IF(ITIPO.EQ.3) JAL(IPOS) = BBPRIM(IPOS)
        GO TO 15
50 CONTINUE

```

Fin del subprograma DERPARG.

```

RETURN
END

```

m) Subprograma FACTBP

Este subprograma tiene la función de factorizar en LU, los modelos de potencia real y reactiva de la isla actual del estudio de flujos.

La forma de acceder este subprograma es mediante la instrucción:

CALL FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAH, JISP, IFACT, NEXT, JL)

cuyos argumentos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NORDEN(NNODOS)	E	ENTERA	Contiene el orden por renglones para efectuar la factorización de BPRIMA y BBPRIM.
JINIC(NNODOS)	E	ENTERA	Indica la posición donde se inicia la información de cada renglón de las matrices BPRIMA y BBPRIM.
JICOL(LONG)	E	ENTERA	Indica la columna asociada al renglón definido por JINIC(.), y que en conjunto marcan la localización de un elemento distinto de cero en BPRIMA y BBPRIM.
JL	E	ENTERA	JL=LONG + elementos creados durante la factorización.
JAH(LONG)	E/S	REAL	Matriz de coeficientes representando a los modelos de potencias activa y reactiva de la isla actual del estudio de flujos.
JISP(LONG)	E	ENTERA	Contiene listas encadenadas de información relevante en cada renglón de BPRIMA y BBPRIM.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
IFACT(NNODOS)	E	ENTERA	Contiene la posición en que inicia la información relevante de cada renglón de BPRIMA y BBPRIM, después de haber efectuado el ordenamiento.
NEXT(JL)	E	ENTERA	Arreglo para encadenar posiciones de elementos distintos de cero en un renglón de la matriz BPRIMA o de la BBPRIM, después de haber efectuado el ordenamiento del sistema de ecuaciones.

El código correspondiente a este subprograma se describe a continuación.

Inicio del subprograma FACTBP.

```
C
SUBROUTINE FACTBP (NNODOS, NORDEN, JINIC, JICOL, JAH, JISP, IFACT, NEXT, JL)
C
```

Declaración de variables.

```

DIMENSION NORDEN(500), JINIC(500), JICOL(2550), JISP(2550),
*          IFACT(500), NEXT(2550)
C
REAL JAH(2550)
C
```

Inicia la factorización.

```

DO 400 I=1, NNODOS
  J = IFACT(I)
  K1 = J
  ID = 0
```

Se recorren el renglón actual hasta antes del elemento que se está factorizando, así como el elemento correspondiente de cada columna para el cálculo de las sumas parciales. K2 indica el elemento del renglón, mientras que L2 indica la columna para efectuar el producto en cada suma parcial.

```

10      IF(J.EQ.0) GO TO 400
        SP1 = 0.0
        K2 = K1
```

```

L4 = JICOL(J)
IF(L4.LT.0) L4 = NORDEN(I)
20      K3 = JICOL(K2)

```

Al cumplirse esta condición se ha llegado al elemento diagonal del renglón que se está factorizando.

```
IF(K2.EQ.J) GO TO 300
```

Esta condición indica que se llegó al elemento diagonal del renglón, pero que el elemento que se está factorizando está localizado en la parte superior de la diagonal de la matriz factorizada.

```

IF(K3.LE.0) GO TO 370
L2 = JINIC(K3)
40      L2 = JISP(L2)
IF(L2.EQ.0) GO TO 100
L3 = JICOL(L2)
IF(L3.NE.L4) GO TO 40

```

En SP1 se almacena la suma parcial de productos de renglón por columna ya factorizados hasta el elemento actual del renglón o la diagonal.

```

SP1 = SP1 + JAH(K2) * JAH(L2)
100     K2 = NEXT(K2)
GO TO 20

```

Este es un elemento de la parte inferior de la diagonal.

```

300     JAH(J) = JAH(J) - SP1
IF(K3.LE.0) GO TO 350
K3 = JINIC(K3)
GO TO 390

```

DIVP es el elemento de la diagonal ya factorizado.

```

350     DIVP = JAH(J)
ID = J
GO TO 390

```

Este es un elemento de la parte superior de la diagonal.

```
370     JAH(J) = (JAH(J) - SP1) / DIVP
```

Se determina el siguiente elemento a factorizar.

```
390     J = NEXT(J)
```

Se regresa a efectuar otra suma de productos.

```

GO TO 10
400 CONTINUE

```

Termina el subprograma FACTBP.

```

RETURN
END

```

n) Subprograma PNODAL

Este subprograma tiene la función de calcular las potencias netas inyectadas en cada nodo de la isla actual del estudio de flujos, utilizando coordenadas rectangulares.

La instrucción para invocar este subprograma es:

```
CALL PNODAL (NNODOS, INIC, ICOL, YBUSR, YBUSI, ISP, VRE, VIM,
*          PCAL, QCAL, NISLAS, INISLA, ISLA, KISLA, ISLACK)
```

cuyos argumentos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
INIC(NNODOS)	E	ENTERA	Indica la posición en que inicia la información de cada renglón de la matriz de admitancias nodal.
ICOL(LONG)	E	ENTERA	Contiene información de la columna asociada con un renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal distinto de cero.
YBUSR(LONG)	E	ENTERA	Parte real de elementos de la matriz de admitancias nodal.
YBUSI(LONG)	E	ENTERA	Parte imaginaria de los elementos de la matriz de admitancias nodal.
ISP(LONG)	E	ENTERA	Lista para encadenar la información relevante de cada renglón de la matriz de admitancias nodal.
VRE(NNODOS)	E	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
VIM(NNODOS)	E	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
PCAL(NNODOS)	E/S	REAL	Potencia real neta inyectada en cada nodo del sistema eléctrico.
QCAL(NNODOS)	E/S	REAL	Potencia reactiva neta inyectada en cada nodo del sistema eléctrico.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
KISLA	E	ENTERA	Identifica a la isla que se le está haciendo el estudio de flujos.
ISLACK(ISLAS)	E	ENTERA	En cada posición contiene el nodo compensador de una isla eléctrica.

El código de este subprograma se describe a continuación.

Inicio del subprograma PNODAL.

```

C
C  SUBROUTINA PARA CALCULAR POTENCIAS NODALES EN LA VERSION RECTANGULAR.
C
C      SUBROUTINE  PNODAL  (NNODOS, INIC, ICOL, YBUSR, YBUSI, ISP, VRE, VIM,
*
*          PCAL, QCAL, NISLAS, INISLA, ISLA, KISLA, ISLACK)
C

```

Declaración de variables.

```

C
C      DIMENSION INIC(550), ICOL(2550), YBUSR(2550), YBUSI(2550),
*
*          ISP(2550), VRE(550), VIM(550), PCAL(550), QCAL(550),
*
*          INISLA(10), ISLA(550), ISLACK(10)
C

```


Se determinan los límites de la lista de localidades contiguas formando los nodos de la isla actual.

```

INICIO = INISLA(KISLA)
IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
IF (KISLA.EQ.NISLAS) IFIN = NNODOS

```

Inicia el cálculo de potencias nodales.

```

DO 90 I1 = INICIO, IFIN
    I = ISLA(I1)

```

Se verifica que el nodo I esté conectado al sistema.

```

IF (I.LT.0) GO TO 90

```

El nodo I está conectado al sistema. Se calculan sus potencias netas inyectadas.

```

IPOS = INIC(I)
AK = 0.0
BK = 0.0

```

Esta condición indica el fin del cálculo de potencias netas inyectadas en el nodo I.

```

70     IF(IPOS.EQ.0) GO TO 75
        GKM = YBUSR(IPOS)
        BKM = YBUSI(IPOS)
        IN = ICOL(IPOS)

```

Esta condición se cumple cuando se está en el elemento diagonal de la matriz de admitancias nodal.

```

IF(IPOS.EQ.INIC(I)) IN = I
AK = AK + GKM * VRE(IN) - BKM * VIM(IN)
BK = BK + GKM * VIM(IN) + BKM * VRE(IN)

```

Se busca el siguiente elemento distinto de cero en la matriz de admitancias nodal.

```

IPOS = ISP(IPOS)
GO TO 70

```

Se almacenan las potencias netas inyectadas en el nodo I.

```

75     PCAL(I) = VRE(I) * AK + VIM(I) * BK
        QCAL(I) = VIM(I) * AK - VRE(I) * BK
90 CONTINUE

```

Fin del subprograma PNODAL.

```

RETURN
END

```

o) Subprograma DESVIA

Este subprograma tiene la finalidad de calcular las desviaciones de potencia en cada nodo, así como determinar la máxima desviación tanto de potencia real como de reactiva y el nodo donde ocurren.

La instrucción para ejecutar este subprograma es:

```
CALL DESVIA (PCAL, QCAL, PG, QG, DMAXP, DMAXQ, NK1, NK2, DELTAP, DELTAQ,
*          NNODOS, TIPO, VESP, VRE, VIM, NPV, NNGEN, NISLAS, INISLA,
*          ISLA, KISLA, ISLACK, NAU, KFUERA, NODFUE)
```

Los argumentos involucrados en este subprograma se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
PCAL(NNODOS)	E	REAL	Potencia real neta inyectada en cada nodo del sistema eléctrico.
QCAL(NNODOS)	E	REAL	Potencia reactiva neta inyectada en cada nodo del sistema eléctrico.
PG(NNODOS)	E	REAL	Potencia real de generación en cada nodo del sistema.
QG(NNODOS)	E	REAL	Potencia reactiva generada en cada nodo del sistema.
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
DMAXP	S	REAL	Máxima desviación de potencia real nodal.
DMAXQ	S	REAL	Máxima desviación de potencia reactiva nodal.
NK1	S	ENTERA	Nodo donde ocurre la máxima desviación de potencia real.
NK2	S	ENTERA	Nodo donde ocurre la máxima desviación de potencia reactiva.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
DELTAQ(NNODOS)	S	REAL	DELTAQ(i) contiene la desviación de potencia real en el nodo i.
DELTAQ(NNODOS)	S	REAL	DELTAQ(i) contiene la desviación de potencia reactiva en el nodo i.
TIPO(NNODOS)	E	ENTERA	Tipo de cada nodo del sistema eléctrico.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema eléctrico.
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.
VRE(NNODOS)	E	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	E	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
NAU(NNGEN)	E	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
KISLA	E	ENTERA	Identifica a la isla que se le está haciendo el estudio de flujos.
ISLACK(ISLAS)	E	ENTERA	En cada posición contiene el nodo compensador de una isla eléctrica.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
VESP(NNGEN)	E	REAL	Voltaje especificado en las terminales de cada nodo de generación.
KFUERA	E	ENTERA	Contador de nodos dados de baja en el subprograma TOPCAM.
NODFUE(KFUERA)	E	ENTERA	Número de cada nodo que ha sido dado de baja en TOPCAM.

El código de este subprograma se describe a continuación.

Inicio del subprograma DESVIA.

C

```

SUBROUTINE DESVIA (PCAL, QCAL, PG, QG, DMAXP, DMAXQ, NK1, NK2, DELTAP,
*                DELTAQ, NNODOS, TIPO, VESP, VRE, VIM, NPV, NNGEN,
*                NISLAS, INISLA, ISLA, KISLA, ISLACK, NAU, KFUERA, NODFUE)

```

Declaración de variables.

```

DIMENSION PCAL(550), QCAL(550), PG(550), QG(550), DELTAP(550),
*          DELTAQ(550), VESP(150), VRE(550), VIM(550), NPV(150),
*          ISLA(550), INISLA(10), ISLACK(10), NAU(150), NODFUE(15)

```

C

```

INTEGER TIPO(550)

```

Se inicializan desviaciones de potencia nodal en cada nodo con ceros.

```

DO 10 I = 1, NNODOS
    DELTAP(I) = 0.0
    DELTAQ(I) = 0.0
10 CONTINUE

```

Se determinan los límites de la lista de localidades contiguas formando los nodos de la isla actual.

```

INICIO = INISLA(KISLA)
IF (KISLA.LT.NISLAS) IFIN = ABS(INISLA(KISLA + 1)) - 1
IF (KISLA.EQ.NISLAS) IFIN = NNODOS

```

NCOMP es el nodo compensador de la isla actual del estudio de flujos.

```

NCOMP = ISLACK(KISLA)

```

Se inicializan las máximas desviaciones de potencia en ceros.

```

DMAXP = 0.0
DMAXQ = 0.0

```

Inicia el cálculo de desviaciones de potencia.

```
DO 90 I1 = INICIO, IFIN
    I = ISLA(I1)
```

Si I es el nodo compensador o está desconectado de la isla actual o del sistema eléctrico, sus desviaciones permanecen en ceros.

```
IF(I.EQ.NCOMP.OR.I.LT.0) GO TO 90
DO 12 J = 1, KFUERA
    IF (I.NE.NODFUE(J)) GO TO 12
    GO TO 90
12    CONTINUE
```

Aquí, se verifica que el nodo no sea un CEV con límites violados. En este caso, la desviación permanece en ceros.

```
DO 15 L = 1, NNGEN
    IF(NPV(L).NE.I) GO TO 15
    IF(NAU(L).GT.0) GO TO 15
    IF(TIPO(I).NE.3) GO TO 15
    GO TO 90
15    CONTINUE
```

Se calcula la desviación de potencia real y se divide entre el voltaje.

```
VMAG = SQRT (VRE(I) * VRE(I) + VIM(I) * VIM(I) )
P1 = ( PG(I) - PCAL(I) ) / VMAG
DELTAP(I) = P1
```

Se verifica si esta desviación en valor absoluto es la máxima.

```
P1 = ABS(P1)
IF(DMAXP.GE.P1) GO TO 20
```

La desviación actual de potencia real es la máxima, por lo que se actualizan DMAXP y NK1.

```
DMAXP = P1
NK1 = I
```

Si el nodo es tipo PV, la desviación de potencia reactiva permanece en cero.

```
20    IF(TIPO(I).NE.3) GO TO 50
```

El nodo es de tipo PQ, por lo que se calcula su desviación de potencia reactiva.

```
Q1 = ( QG(I) - QCAL(I) ) / VMAG
GO TO 60
50    Q1 = 0.0
60    DELTAQ(I) = Q1
```

Se verifica si esta desviación en valor absoluto es la máxima.

```
Q1 = ABS(Q1)
```

```
IF(DMAXQ.GE.Q1) GO TO 90
```

La desviación actual de potencia reactiva es la máxima, por lo que se actualizan DMAXQ y NK2.

```
DMAXQ = Q1
```

```
NK2 = I
```

```
90 CONTINUE
```

Fin del subprograma DESVIA.

```
RETURN
```

```
END
```

p) Subprograma SOLBBP

La función de este subprograma es la resolución de los sistemas de ecuaciones lineales de los modelos de potencia activa y reactiva en cada iteración. Para ésto, tanto BPRIMA(.), como BBPRIM(.) ya llegan factorizados.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL SOLBBP (NNODOS, NORDEN, JICOL, JAH, DELTAP, IFACT, NEXT, JL)
```

cuyos argumentos se describen a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
NORDEN(NNODOS)	E	ENTERA	Contiene el orden por renglones para efectuar la factorización de BPRIMA y BBPRIM.
JICOL(LONG)	E	ENTERA	Indica la columna asociada al renglón definido por JINIC(.), y que en conjunto marcan la localización de un elemento distinto de cero en BPRIMA y BBPRIM.
JL	E	ENTERA	JL=LONG + elementos creados durante la factorización.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
JAH(LONG)	E	REAL	Matriz de coeficientes representando a los modelos de potencias activa y reactiva de la isla actual del estudio de flujos.
DELTAP(NNODOS)	E/S	REAL	DELTAP(i) contiene la desviación de potencia real en el nodo i, al entrar a este subprograma. Al salir contiene ya sea incrementos de ángulo o de magnitud de voltaje.
IFACT(NNODOS)	E	ENTERA	Contiene la posición en que inicia la información relevante de cada renglón de BPRIMA y BBPRIM, después de haber efectuado el ordenamiento.
NEXT(JL)	E	ENTERA	Arreglo para encadenar posiciones de elementos distintos de cero en un renglón de la matriz BPRIMA o de la BBPRIM, después de haber efectuado el ordenamiento del sistema de ecuaciones.

El código de este subprograma se describe a continuación.

Inicio del subprograma SOLBBP.

```

C
  SUBROUTINE SOLBBP (NNODOS, NORDEN, JICOL, JAH, DELTAP, IFACT, NEXT, JL)
C
  DIMENSION NORDEN(550), IFACT(550), NEXT(2550), DELTAP(550), JICOL(2550)
C
  REAL JAH(2550)
C
C SE REALIZA SUSTITUCION HACIA ADELANTE.
C
  DO 500 I = 1, NNODOS

```

N1 corresponde a la variable para la cual se está resolviendo.

```

  N1 = NORDEN(I)
  SP = 0.0

```

```

      J = IFACT(I)
410    L4 = JICOL(J)

```

Al cumplirse esta condición, se tiene el valor de la variable actual, es decir, se ha obtenido el valor de tal variable mediante substitución hacia adelante. Obviamente, esta variable no es la de estado, sino una intermedia, la cual sirve para calcular la de estado.

```

      IF(L4.LE.0) GO TO 450
      SP = SP + JAH(J) * DELTAP(L4)
      J = NEXT(J)
      GO TO 410

```

Este es el valor de la variable intermedia.

```

450    DELTAP(N1) = (DELTAP(N1) - SP) / JAH(J)
500 CONTINUE

```

```

C
C    SE REALIZA SUSTITUCION HACIA ATRAS.
C

```

Como los comentarios indican, se inicia el proceso por la última variable de estado, terminando por la primera.

```

      DO 600 I = NNODOS, 1, - 1
          N1 = NORDEN(I)
          SP = 0.0
          J = IFACT(I)
520    L4 = JICOL(J)

```

Al cumplirse esta condición, se está en la parte superior de la matriz factorizada y entonces se hace la substitución hacia atrás.

```

      IF(L4.LE.0)GO TO 540

```

Se está en la parte inferior de la matriz factorizada y, entonces, se busca otro elemento que esté localizado en la parte superior.

```

          J = NEXT(J)
          GO TO 520
540    K1 = J
550    J = NEXT(J)
          IF(J.EQ.0) GO TO 560
          L4 = JICOL(J)
          SP = SP + JAH(J) * DELTAP(L4)
          GO TO 550

```

Al cumplirse esta condición, se tiene el valor de la variable actual.

```

560    DELTAP(N1) = DELTAP(N1) - SP
600 CONTINUE

```

Fin del subprograma SOLBBP.

```

      RETURN
      END

```


g) Subprograma CORRIG

Este subprograma tiene como función actualizar las variables de estado (ángulos y voltajes), después de haber calculado las soluciones de los sistemas ecuaciones lineales de los modelos de potencia activa y reactiva, para la iteración actual.

La instrucción para invocar este subprograma es:

```
CALL CORRIG (NNODOS, VRE, VIM, VOLT, ANGULO, DELTAP, DELTAQ, KA)
```

cuya lista de argumentos se describe a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
VRE(NNODOS)	E/S	REAL	Parte real del voltaje complejo en cada nodo del sistema eléctrico.
VIM(NNODOS)	E/S	REAL	Parte imaginaria del voltaje complejo en cada nodo del sistema eléctrico.
VOLT(NNODOS)	E/S	REAL	Magnitud del voltaje complejo en cada nodo del sistema eléctrico.
ANGULO(NNODOS)	E/S	REAL	Angulo de fase en radianes del voltaje complejo nodal en el sistema eléctrico.
DELTAP(NNODOS)	E	REAL	Incremento en ángulos al haber resuelto el modelo de potencia activa en la iteración actual.
DELTAQ(NNODOS)	E	REAL	Incremento en magnitudes de voltaje, al haber resuelto el modelo de pot. reactiva en la iteración actual.
KA	E	ENTERA	Indica si la actualización es para ángulos o magnitudes de voltaje.

El código del subprograma se muestra a continuación.

Inicio del subprograma CORRIG.

```
C
  SUBROUTINE CORRIG (NNODOS, VRE, VIM, VOLT, ANGULO, DELTAP, DELTAQ, KA)
C
```

Declaración de variables.

```

  DIMENSION VOLT(550), ANGULO(550), VRE(550), VIM(550),
  *          DELTAP(550), DELTAQ(550)
C
```

Si KA = 0, la actualización de variables es solamente en magnitudes de voltaje.

```
  IF (KA.EQ.0) GO TO 2
```

Actualización de ángulos de fase.

```

  DO 1 I = 1, NNODOS
    ANGULO(I) = ANGULO(I) + DELTAP(I)
  1 CONTINUE
  GO TO 4
```

Actualización de magnitudes de voltaje.

```

  2 DO 3 I = 1, NNODOS
    VOLT(I) = VOLT(I) + DELTAQ(I) * VOLT(I)
  3 CONTINUE
C
```

Se actualiza la parte real y la imaginaria de los voltajes complejos.

```

  4 DO 5 I = 1, NNODOS
    VRE(I) = VOLT(I) * COS(ANGULO(I))
    VIM(I) = VOLT(I) * SIN(ANGULO(I))
  5 CONTINUE
C
```

Fin del subprograma CORRIG.

```

  RETURN
  END
```

r) Subprograma REVNPV

Este subprograma tiene como función revisar los límites de potencia reactiva en cada nodo de generación (tipo voltaje controlado o PV), así como actualizar las variables de cada nodo de acuerdo a un cambio de tipo, ya sea de PV a PQ o viceversa.

La forma de ejecutar este subprograma es mediante la instrucción:

```
CALL REVNPV (NNODOS, TIPO, NNGEN, NPV, QMIN, QMAX, VESP, INPV,
*           QG, QC, QCAL, DELTAQ, VOLT, KVC, NISLAS, INISLA,
*           ISLA, KISLA, NAU, NCEVS, NCEVV, YM, INIC, ICOL,
*           YBUSI, ISP, DELTAP, NLT, BPRIMA, BBPRIM, ISLACK)
```

cuya lista de argumentos es descrita a continuación.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NNODOS	E	ENTERA	Número total de nodos en el sistema eléctrico.
TIPO(NNODOS)	E/S	ENTERA	Tipo de cada nodo del sistema eléctrico.
NNGEN	E	ENTERA	Número total de nodos de generación en el sistema eléctrico.
NPV(NNGEN)	E	ENTERA	Contiene el número de cada nodo de generación.
QMIN(NNGEN)	E/S	REAL	Límite inferior de potencia reactiva en cada nodo de generación del sistema.
QMAX(NNGEN)	E/S	REAL	Límite superior de potencia reactiva en cada nodo de generación del sistema.
VESP(NNGEN)	E	REAL	Voltaje especificado en las terminales de cada nodo de generación.
INPV(NNGEN)	E/S	ENTERA	Indicador del estado de los límites de potencia reactiva generada.
QG(NNODOS)	E/S	REAL	Potencia reactiva generada en cada nodo del sistema.

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
QC(NNODOS)	E	REAL	Potencia reactiva de carga en cada nodo del sistema.
QCAL(NNODOS)	E	REAL	Potencia reactiva neta inyectada en cada nodo del sistema.
DELTAQ(NNODOS)	E/S	REAL	Desviaciones de potencia reactiva en cada nodo del sistema eléctrico.
DELTAP(NNODOS)	E/S	REAL	Desviaciones de potencia activa en cada nodo del sistema eléctrico.
VOLT(NNODOS)	E/S	REAL	Magnitud del voltaje complejo en cada nodo del sistema eléctrico.
KVC	S	ENTERA	Indica si se han detectado violaciones de límites de potencia reactiva generada, tal que al salir del subprograma se tomen las acciones necesarias en SOLFLU, que es el subprograma que invocó a REVNPV.
NISLAS	E	ENTERA	Total de islas eléctricas en el sistema.
INISLA(NISLAS)	E	ENTERA	INISLA(i) indica posición en la cual inicia la lista encadenada correspondiente a la isla i.
ISLA(NNODOS)	E	ENTERA	Lista de encadenamiento que define los nodos que forman cada isla.
KISLA	E	ENTERA	Identifica a la isla que se le está haciendo el estudio de flujos.
NAU(NNGEN)	E/S	ENTERA	Nodo de alta tensión que está asociado al nodo generador indicado en la misma posición del arreglo NPV(.).

NOMBRE	ENTRADA O SALIDA	TIPO	DESCRIPCION
NCEVS	E	ENTERA	Total de compensadores estáticos de potencia reactiva.
NCEVV(NCEVS)	E/S	ENTERA	Indica el nodo de voltaje controlado tipo CEV, cuyos límites de potencia reactiva han sido violados.
YM(NCEVS)	E/S	REAL	Susceptancia capacitiva o inductiva equivalente a los límites violados de un compensador estático de potencia reactiva.
INIC(NNODOS)	E	ENTERA	Indica la posición en que inicia la información de cada renglón de la matriz de admitancias nodal.
ICOL(LONG)	E	ENTERA	Contiene información de la columna asociada con un renglón definido por INIC(.), a la cual corresponde un elemento de la matriz de admitancias nodal distinto de cero.
YBUSI(LONG)	E/S	ENTERA	Parte imaginaria de los elementos de la matriz de admitancias nodal.
ISP(LONG)	E	ENTERA	Lista para encadenar la información relevante de cada renglón de la matriz de admitancias nodal.
NLT(NNODOS)	E/S	ENTERA	Indica si el nodo asociado a cada posición contiene compensación fija en derivación. En este caso, $NLT(i) = 1$, donde i es el nodo asociado.
ISLACK(ISLAS)	E	ENTERA	En cada posición contiene el nodo compensador de una isla eléctrica.

El código de este subprograma se describe a continuación.

Inicio del subprograma REVNPV.

```
C
C   SUBROUTINA PARA REVISAR LIMITES DE POTENCIA REACTIVA GENERADA EN
C   NODOS DE VOLTAJE CONTROLADO (P-V).
C
C   SUBROUTINE REVNPV (NNODOS, TIPO, NNGEN, NPV, QMIN, QMAX, VESP, INPV,
*   QG, QC, QCAL, DELTAQ, VOLT, KVC, NISLAS, INISLA,
*   ISLA, KISLA, NAU, NCEVS, NCEVV, YM, INIC, ICOL,
*   YBUSI, ISP, DELTAP, NLT, BPRIMA, BBPRIM, ISLACK)
```

Declaración de variables.

```

DIMENSION NPV(150), QMIN(150), QMAX(150), VESP(150), INPV(150),
*   QG(550), QC(550), QCAL(550), DELTAQ(550), INIC(550),
*   VOLT(550), NCEVV(150), ISP(2550), ICOL(2550), YM(150),
*   YBUSI(2550), DELTAP(550), INISLA(10), BBPRIM(2550),
*   NLT(550), NAU(150), BPRIMA(2550), ISLA(550), ISLACK(10)
C
C   INTEGER TIPO(550)
C
```

Se determinan los límites de la lista de localidades contiguas formando los nodos de la isla actual.

```

K = INISLA(KISLA)
IF(KISLA.LT.NISLAS) K2 = ABS(INISLA(KISLA + 1)) - 1
IF(KISLA.EQ.NISLAS) K2 = NNODOS
```

Se efectúa revisión del estado de nodos PV, es decir, con límites sin haber sido violados en iteraciones anteriores, o si tiene límites violados en iteraciones anteriores.

```
DO 200 J = 1, NNGEN
```

I es el nodo generador que se revisa.

```
I = NPV(J)
```

Esta condición se refiere a la posibilidad de que el generador esté desconectado.

```
IF (I.LT.0) GO TO 200
```

Se verifica que el nodo que se está revisando no sea el compensador de la isla actual.

```

DO 15 I1 = 1, NISLAS
   IF (INISLA(I1).LT.0) GO TO 15
   IF (I.EQ.ISLACK(I1)) GO TO 200
15   CONTINUE
```

Se verifica que el nodo I esté en la isla actual del estudio de flujos.

```
DO I1 = K, K2
```

```

      IF (I.EQ.ISLA(I1)) GO TO 16
    END DO

```

El nodo de generación I no está en la isla actual del estudio de flujos

```

      GO TO 200

```

Se observa el estado actual de nodo I.

```

16      V1 = VOLT(I)
        VE = VESP(J)
        I1 = INPV(J)

```

Esta condición se refiere a si el nodo es una máquina síncrona o CEV.

```

      IF (NAU(J).LT.0) GO TO 60

```

Se trata de una máquina síncrona. Si $I1 = 0$, entonces I, hasta este momento está generando potencia reactiva dentro de límites, los cuales serán revisados inmediatamente.

```

      IF (I1.EQ.0)      GO TO 30

```

Se trata de una máquina síncrona. Si $I1 < 0$, entonces I tiene el límite inferior violado desde iteraciones anteriores.

```

      IF (I1.LT.0)      GO TO 20

```

Se trata de una máquina síncrona. Si $I1 > 0$, entonces I tiene el límite superior violado desde iteraciones anteriores.

```

      IF (V1.GE.VE)      GO TO 25

```

En este caso de límite superior violado, si no es mayor o igual la magnitud de voltaje actual que la especificada, el nodo continuará como tipo PQ.

```

      GO TO 200
20      IF(V1.GE.VE) GO TO 50

```

El nodo regresa a ser tipo PV, después de haber tenido el límite superior violado desde iteraciones anteriores.

```

25      TIPO(I) = 2
        VOLT(I) = VE
        INPV(J) = 0
        KVC = 1
        GO TO 200

```

Se revisa el límite superior de un nodo tipo PV.

```

30      Q1 = QCAL(I)
        IF(Q1.LE.QMAX(J)) GO TO 35

```

Se detecta violación del límite superior, por lo que se cambia el tipo a PQ, se especifica a QG(I) con el límite máximo y se indica este movimiento dando un valor a INPV(J) de 1.

```

QG(I) = QMAX(J)
INPV(J) = 1
GO TO 40

```

Se revisa el límite inferior de un nodo PV.

```

35      IF (Q1.GE.QMIN(J)) GO TO 200

```

Se detecta violación del límite inferior, por lo que se cambia el tipo a PQ, se especifica a QG(I) con el límite mínimo y se indica este movimiento dando un valor a INPV(J) de -1.

```

40      QG(I)      = QMIN(J)
        INPV(J)   = - 1
        TIPO(I)   = 3

```

Se calcula la nueva desviación de potencia reactiva en el nodo I.

```

        DELTAQ(I) = QG(I) - QCAL(I)

```

Se actualiza el indicador de cambio de tipo de nodo para notificar ésto en SOLFLU.

```

50      KVC = 1
        GO TO 200

```

Se trata de un compensador estático de VARs. Se verifica que su estado sea el de un nodo PV.

```

60      IF (I1.EQ.0) GO TO 90

```

Se trata de un compensador estático de potencia reactiva con un límite violado. En este caso, no se verifica la posibilidad de que regrese a ser un nodo PV.

```

        GO TO 200

```

Se revisan los límites inferior y superior del CEV.

```

90      Q1 = QCAL(I)
        IF (Q1.LE.QMAX(J)) GO TO 100

```

Se detecta violación del límite superior, por lo que se cambia el tipo a PQ, se especifica a QG(I) con el límite máximo y se indica este movimiento dando un valor a INPV(J) de 1. Además, se calcula la admitancia equivalente al límite violado que se conectará en derivación al nodo de alta tensión asociado.

```

100     QG(I)      = QMAX(J)
        INPV(J)   = 1
        Y1 = QMAX(J) / (VE * VE)
        GO TO 120
        IF(Q1.GE.QMIN(J))GO TO 200

```


Se detecta violación del límite inferior, por lo que se cambia el tipo a PQ, se especifica a QG(I) con el límite mínimo y se indica este movimiento dando un valor a INPV(J) de -1. Además, se calcula la admitancia equivalente al límite violado que se conectará en derivación al nodo de alta tensión asociado.

```

QG(I) = QMIN(J)
INPV(J) = - 1
Y1 = QMIN(J) / (VE * VE)

```

Se efectúa el cambio de tipo de nodo. Se actualizan las desviaciones de potencia activa y reactiva, se actualiza KVC, se llenan los arreglos NCEVV(.) conteniendo al nodo con el límite violado y YM(.) conteniendo la admitancia equivalente al mismo. Por último, se modifica la matriz de admitancias nodal en el elemento diagonal del nodo de alta tensión asociado al CEV con su límite violado, así como el fuera de la misma, para simular la desconexión del CEV del sistema eléctrico.

```

120      TIPO(I) = 3
          DELTAQ(I) = 0.0
          DELTAP(I) = 0.0
          KVC = 2
          NCEVV(J) = I
          YM(J) = Y1
          K = INIC(I)
160      K = ISP(K)
          IF (K.EQ.0) GO TO 200
          JK = ICOL(K)
          IK = INIC(JK)
          K1 = IK
170      IK = ISP(IK)
          IF (IK.EQ.0) GO TO 200
          IC = ICOL(IK)
          IF (IC.EQ.1) GO TO 175
          GO TO 170
175      YBUSI(K1) = Y1 + YBUSI(IK) + YBUSI(K1)

```

YBUSI(IK) = 0, debido a que se está simulando la desconexión del CEV del sistema eléctrico de potencia.

```

YBUSI(IK) = 0.0

```

Se actualiza el valor de NLT(.) para indicar que existe un compensador fijo en derivación conectado al nodo de alta tensión asociado al CEV con su límite violado.

```

          NLT(JK) = 1
200 CONTINUE

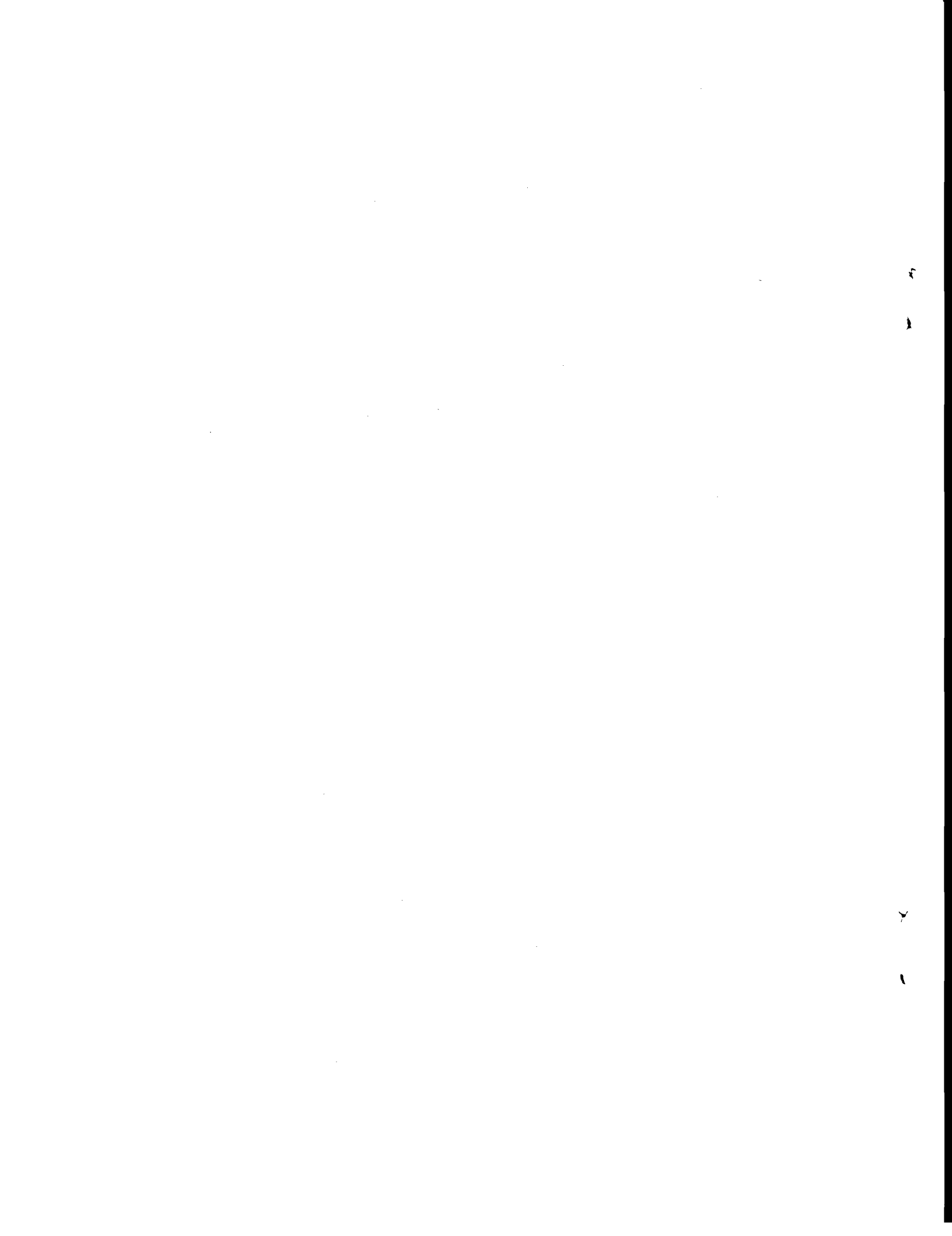
```

Fin del subprograma REVNPV.

```

RETURN
END

```



8. Referencias

1. R. Mota Palomino, **Programa de Flujos de Potencia, Vol. II: Manual de Programa y Código Fuente**, Comisión de Estudios para América Latina y el Caribe (CEPAL), LC/MEX/R.198, Enero de 1990.
2. R. Mota Palomino, **Técnicas Computacionales Eficientes para el Análisis de Sistemas Eléctricos de Potencia**, Tesis de Maestría, Sección de Posgrado e Investigación ESIME-IPN, México, D.F., 1979.

2

3

4

5

1

2

3

4

1

1

1

1