

**ESCUELA SUPERIOR
POLITECNICA DEL LITORAL**

FACULTAD DE INGENIERIA ELECTRICA

**" COORDINACION DE LA PROTECCION
DE LINEAS DE DISTRIBUCION POR
MEDIO DE RELES DE
SOBRECORRIENTE UTILIZANDO UN
PROGRAMA PARA
MICROCOMPUTADORES "**

TESIS DE GRADO

PREVIA A LA OBTENCION DEL TITULO DE :

**INGENIERO EN ELECTRICIDAD
ESPECIALIZACION : POTENCIA**

PRESENTADA POR :

EGBERTO RAMOS ROJAS

GUAYAQUIL - ECUADOR

1987

AGRADECIMIENTO

AL ING. JORGE CHIRIBOGA, Director de Tesis.

A mis compañeros de estudios, por su ayuda, sin la cual no se hubiera podido llevar a cabo este trabajo.

A todos mis profesores de la Facultad, por sus conocimientos compartidos.

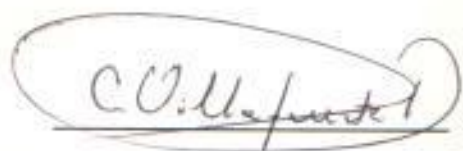
DEDICATORIA

A MIS PADRES

A MIS HERMANAS

A VERONICA

A MIS AMIGOS

A handwritten signature in black ink, enclosed in a hand-drawn oval. The signature appears to be 'C. Villafuerte'.

ING. CARLOS VILLAFUERTE
Subdecano de la F.I.E.

A handwritten signature in black ink, enclosed in a hand-drawn oval. The signature appears to be 'J. Chiriboga'.

ING. JORGE CHIRIBOGA
Director de Tesis

A handwritten signature in black ink, enclosed in a hand-drawn oval. The signature appears to be 'Gustavo Bermudez'.

ING. GUSTAVO BERMUDEZ
Miembro del Tribunal

A handwritten signature in black ink, enclosed in a hand-drawn oval. The signature appears to be 'A. Hanze'.

ING. ALBERTO HANZE
Miembro del Tribunal

DECLARACION EXPRESA

" La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me corresponden exclusivamente ; y el patrimonio intelectual de la misma, a la **ESCUELA SUPERIOR POLITECNICA DEL LITORAL** ".

(Reglamento de Exámenes y Títulos profesionales de la ESPOL).



.....

EGBERTO RAMOS ROJAS

RESUMEN

El presente trabajo consiste en lograr una herramienta muy práctica para la coordinación de la protección de líneas de distribución con relevadores de sobrecorriente y de líneas de transmisión de energía eléctrica en general.

La tesis consiste de un estudio teórico primario para luego entrar a la programación y a la descripción de las opciones, ventajas y desventajas del programa.

El programa de coordinación consta de tres módulos principales : La base de datos, la hoja de coordinación y el archivo tutor. La base de datos nos permite almacenar en parejas de puntos las curvas de los relevadores ; La hoja de coordinación permite la graficación de las curvas de la base de datos ; El Archivo tutor nos relata las opciones del programa.

INDICE GENERAL

RESUMEN	6
INDICE GENERAL	7
INDICE DE FIGURAS	11
INDICE DE ABREVIATURAS	12
INTRODUCCION	13

CAPITULO I

PROTECCION DE LINEAS CON RELEVADORES DE SOBRECORRIENTE	14
---	----

1.1 GENERALIDADES

1.1.1 CLASIFICACION DE LAS LINEAS DE ENERGIA ELECTRICA	14
1.1.2 TECNICAS PARA LA PROTECCION DE LINEAS DE TRANSMISION	16

1.1.3	SISTEMAS DE PROTECCION CON RELEVADORES PARA FALLAS EN LINEAS DE TRANSMISION	17
1.1.4	FACTORES A CONSIDERAR PARA LA SELECCION DEL SISTEMA DE PROTECCION DE UNA LINEA DE TRANSMISION	19
1.2	PROTECCION DE LINEAS DE CIRCUITOS DE DISTRIBUCION CON RELEVADORES DE SOBRECORRIENTE	20
1.2.1	APLICACION DE RELEVADORES EN ALIMENTADORAS RADIALES	21
1.2.2	COORDINACION	25
1.2.3	INTERVALO DE TIEMPO DE COORDINACION (CTI)	30
1.2.4	SELECCION DE LA TOMA DE CORRIENTE (TAP) DE UN RELEVADOR DE SOBRECORRIENTE	31
1.2.5	SELECCION DE LA CURVA DE SOBRECORRIENTE DE TIEMPO DEL RELEVADOR	33
1.2.6	APLICACIONES DEL DISPARO INSTANTANEO	40
1.2.7	RELEVADORES DE FASE Y DE TIERRA	41
1.2.8	COORDINACION FUSIBLE-RELEVADOR	42
1.2.9	COORDINACION CON RECONECTADORES Y SECCIONALIZADORES	45
1.2.10	SISTEMAS DE PROTECCION PARA ALIMENTADORAS DE DISTRIBUCION	47

CAPITULO II

PROGRAMA DE COORDINACION UTILIZANDO RELEVADORES DE

SOBRECORRIENTE	52
2.1 INTRODUCCION	52
2.2 LENGUAJE DE PROGRAMACION UTILIZADO	53
2.3 PRESENTACION Y MENU PRINCIPAL	54
2.4 BASE DE DATOS.- INTRODUCCION	56
2.4.1 MENU DE LA BASE DE DATOS	56
2.4.2 GENERALIDADES	58
2.4.3 OPCION CREAR CURVA	59
2.4.4 OPCION ELIMINAR CURVA	60
2.4.5 OPCION EDITAR CURVA	61
2.4.6 SALIR AL MENU PRINCIPAL	62
2.5 HOJA DE COORDINACION.- INTRODUCCION	62
2.5.1 MENU DE LA HOJA DE COORDINACION	62
2.5.2 GENERALIDADES	
2.5.3 OPCION HABILITAR CURVA	65
2.5.4 OPCION DIBUJAR CURVA	66
2.5.5 OPCION ELIMINAR CURVA	67
2.5.6 OPCION DESPLAZAR EN TIEMPO Y CORRIENTE	67
2.5.7 VALOR DE TIEMPO Y VALOR DE CORRIENTE	68

2.5.8	COORDINACION	70
2.5.9	IMPRIMIR	71
2.5.10	SALIR AL MENU PRINCIPAL	72
2.6	ARCHIVO TUTOR	72
2.7	SALIR AL D.O.S.	73
	CONCLUSIONES Y RECOMENDACIONES	74

APENDICES

A.-	DIAGRAMA DE BLOQUES DEL PROGRAMA	76
B.-	CORRIDA EJEMPLO	80
C.-	LISTADO DEL PROGRAMA	89

BIBLIOGRAFIA	130
---------------------------	-----

INDICE DE FIGURAS

No.		PAG.
1	ALIMENTADORA RADIAL TIPICA	23
2.A	PROCEDIMIENTO DE AJUSTE DE COORDINACION PARA LOS RELEVADORES DEL DISYUNTOR "G" DE LA FIG. 1 (Corr. Máximas) ...	27
2.B	PROCEDIMIENTO DE AJUSTE DE COORDINACION PARA LOS RELEVADORES DEL DISYUNTOR "G" DE LA FIG. 1 (Corr. Mínimas) ...	28
3	COMPARACION DE LAS CARACTERISTICAS DE SOBRE CORRIENTE DEL TIPO (CO)	35
4	COMPARACION ENTRE LOS RELEVADORES DE SOBRE CORRIENTE DE TIEMPO FIJO VS. TIEMPO INVERSO EN CIRCUITOS CON ALIMENTADORAS RADIALES	37
5	COMPARACION ENTRE LAS CURVAS DE MAGNITUD DE CORRIENTE (ESCALA LINEAL) DE FUSIBLES Y RELEVADORES	43
6	PROTECCION TIPICA DE UNA ALIMENTADORA DE DISTRIBUCION	48
7	CURVAS TIPICAS DE COORDINACION PARA LOS DISPOSITIVOS DE PROTECCION DE LA FIG. 6	50
8	MENU Y PRESENTACION DEL PROGRAMA	55
9	MENU Y PRESENTACION DE LA BASE DE DATOS	57
10	MENU Y PRESENTACION DE LA HOJA DE COORDINACION	64
11	EJEMPLO.- PROTECCION DE LINEAS RADIALES ADYACENTES	
12	EJEMPLO 2.- PROTECC. DE RESPALDO PARA SIST. DE DIST.	

INDICE DE ABREVIATURAS

C.O.	:	CARACTERISTICA DE SOBRE CORRIENTE
CTI	:	INTERVALO DE TIEMPO DE COORDINACION
D.C.	:	CORRIENTE DIRECTA
E.H.V.	:	EXTRA ALTO VOLTAJE
H.V.	:	ALTO VOLTAJE
I_{pK}	:	CORRIENTE DE ARRANQUE DEL RELEVADOR
K.V.	:	KILO VOLTIOS
L.T.	:	LINEA DE TRANSMISION
P.U.	:	POR UNIDAD
S.T.M.	:	TIEMPO DE DURACION DE MAX. CARGA (CORTO)
TAP	:	TOMA DE CORRIENTE DEL RELEVADOR
V	:	VOLTIOS
U.H.V.	:	Ultra Alto Voltaje
D.O.S.	:	Sistema Operativo de la computadora IBM
PICK-UP	:	Corriente de arranque de los relevadores.

INTRODUCCION

El arte de las protecciones es tarea muy confortante, pero un poco difícil al no contar con las herramientas necesarias.

Al contar con una herramienta como un programa de computadoras que almacene las curvas de los dispositivos de protección a utilizarse en los sistemas de potencia eléctrica y posteriormente permita graficarlas y manejarlas, y más aún que nos diga si para una determinada corriente de falla, nuestro escogitamiento y ajuste de los dispositivos de protección son los adecuados, la protección de sistemas de potencia y la selectividad no serán más un trabajo tedioso. Por el contrario, el arte de las protecciones se convertirá en un trabajo versátil y ameno.

CAPITULO I

PROTECCION DE LINEAS CON RELEVADORES DE SOBRECORRIENTE

1.1 INTRODUCCION

La mayoría de las fallas experimentadas en un sistema de potencia ocurren en las líneas conectadas a fuentes de generación.

Estos circuitos varían ampliamente en sus características, longitudes, configuraciones, e importancia relativa, así varían también sus esquemas de protección.

1.1.1 CLASIFICACION DE LAS LINEAS DE ENERGIA ELECTRICA

Las líneas de energía eléctrica se clasifican generalmente por su función, la cual está relacionada con su nivel de voltaje. La clasificación típica es la siguiente:

a. **Distribución** (2.4 KV - 34.5 KV) : Circuitos

transmitiendo energía hasta los tomacorrientes del usuario.

- b. **Subtransmisión** (13.8 KV - 138 KV) : Circuitos transmitiendo energía a las subestaciones de distribución.
- c. **Transmisión** (69 KV - 765 KV) : Circuitos transmitiendo energía entre subestaciones mayores o interconectando sistemas .

Las líneas de transmisión se dividen a su vez en :

- 1. **Alto voltaje** (HV) : 115 KV -230 KV
- 2. **Extra alto voltaje** (EHV) : 345 KV -765 KV
- 3. **Ultra alto voltaje** (UHV) : mayores que 765 KV

Los sistemas de corriente directa (D.C.) pueden ser clasificados de la siguiente manera :

- a. **Bajo voltaje** (240 V -250 V) : Energía auxiliar en plantas de energía y subestaciones ; circuitos de control; y , ocasionalmente en algunas plantas industriales.
- b. **Medio voltaje** (300 V - 600 V) : Industria de la

transportación.

- c. **Alto voltaje** (mayores que 600 V) : Transmisión a larga distancia y sistemas grandes de interconexión.

1.1.2 TECNICAS PARA LA PROTECCION DE LINEAS DE TRANSMISION

Existen siete técnicas de protección con relevadores comunmente utilizadas para despejar fallas en líneas de transmisión :

- a. **Sobre corriente instantánea**
- b. **Sobre corriente de tiempo inverso**
- c. **Direccional instantánea y/o sobre corriente de tiempo inverso**
- d. **Sobre corriente por pasos** (Step time overcurrent)
- e. **Protección de distancia de tiempo inverso** (Inverse time-distance)
- f. **Protección de distancia por zona**

g. Protección piloto

1.1.3 SISTEMAS DE PROTECCION CON RELEVADORES PARA FALLAS EN LINEAS

Los sistemas de relevadores para protección de fallas de fase de líneas de transmisión están descritos en la Tabla 1 (según Westinghouse). La última columna en cada tabla nos da, con relativa proximidad, el costo por unidad de los sistemas de equipos de relevadores del sistema básico de sobrecorriente de tiempo inverso (CO) .

Tipo de protección	Tipo básico de relevador	Costo Relativo (pu)
Sobrecorriente / tiempo	CO	1,00
Instantánea y sobre corriente de tiempo	CO con HT	1,30
Sobrecorriente de tiempo direccional	CR	2,70
Instant. y sob. corriente de tiempo direccional	CR con HT	2,90
Sobrecorriente por paso	CO-4	3,80
Tiempo inverso a dist.	KD	6,75
Distancia por Zona	KD y TD4	10,20

Tabla 1. Sistemas de relevadores de protección (Westinghose)

1.1.4 FACTORES A CONSIDERAR PARA LA SELECCION DEL SISTEMA DE PROTECCION DE UNA LINEA DE TRANSMISION

Varios factores fundamentales influyen en la selección final de la protección aplicada a una línea de transmisión:

- a. **Tipo de circuito** : cable, poste, línea simple, líneas paralelas, multi-terminal, etc.
- b. **Función de la línea e importancia** : efecto sobre la continuidad en el servicio, requerimientos de tiempos prácticos y reales para aislar la falla del resto del sistema.
- c. **Coordinación con el sistema existente** : compatibilidad con el equipo en las líneas y sistemas asociados.

A estas tres consideraciones se deben añadir factores económicos y la preferencia de los ingenieros por determinados relevadores, basada en los conocimientos técnicos y la experiencia. Ya que hay

muchas consideraciones, no es posible establecer reglas fijas para la protección de líneas de transmisión. En el resto de este capítulo, sin embargo, se enfocarán reglas básicas de aplicación y procedimientos de coordinación para ayudar al ingeniero en la selección propia del sistema de protección para circuitos de distribución, para fallas de fase .

1.2 PROTECCION DE LINEAS DE CIRCUITOS DE DISTRIBUCION CON RELEVADORES DE SOBRECORRIENTE

1.2.1 INTRODUCCION

Las líneas que salen de las subestaciones de distribución son usualmente radiales a la carga del área. Lazos abiertos, operando normalmente como líneas radiales, proporcionan alternativas de suministro a la carga desde diferentes subestaciones. Estos lazos pueden estar temporalmente cerrados para que la energía fluya entre las subestaciones;

en cualquier dirección de la línea.

Para el propósito de la aplicación de relevadores, se considera una alimentadora como radial si, en una localización particular del relevador, la máxima corriente de falla en la dirección del no-disparo (back-feed) es menor que el veinticinco por ciento (25%) de la corriente mínima de falla para la cual el relevador de protección opera.

Los siguientes temas cubren los sistemas de protección para circuitos radiales de distribución .

1.22 APLICACION DE RELEVADORES EN ALIMENTADORAS RADIALES

Las alimentadoras radiales pueden ser protegidas por relevadores de sobrecorriente no-direccionales. La FIG. 1 muestra varias secciones de una alimentadora radial típica. A causa de que la alimentadora es radial, cada sección requiere sólo un disyuntor (breaker) en la fuente o en el final de la subestación de distribución. Para despejar las fallas en (1) y las otras fallas a la derecha , solamente el disyuntor (breaker) en R

necesita dispararse. Para despejar las fallas en (2) y (3) y en el área entre ellos, el disyuntor en H debe dispararse. De la misma manera, para despejar las fallas en (4) y (5) y entre ellos, el disyuntor en G debe dispararse.

Ninguno de los relevadores en la localización de los disyuntores puede distinguir si la falla remota es en la línea protegida, en una barra lejana, o en una línea adyacente. Los relevadores en H, por ejemplo, no pueden distinguir entre las fallas en (1) o en (2), ya que la magnitud de la corriente medida en H será la misma en ambos casos. No es deseable que el disyuntor H se abra para fallas en (1), ya que esto interrumpiría innecesariamente la carga entre H y R. Existen dos técnicas disponibles para resolver este problema: retardo de tiempo (**time delay**) y relevador piloto.

Los relevadores de tiempo retardan la operación del relevador para una falla remota, permitiendo a los relevadores y disyuntores cercanos a la falla, despejarla si es posible. En el ejemplo de la FIG. 1, los

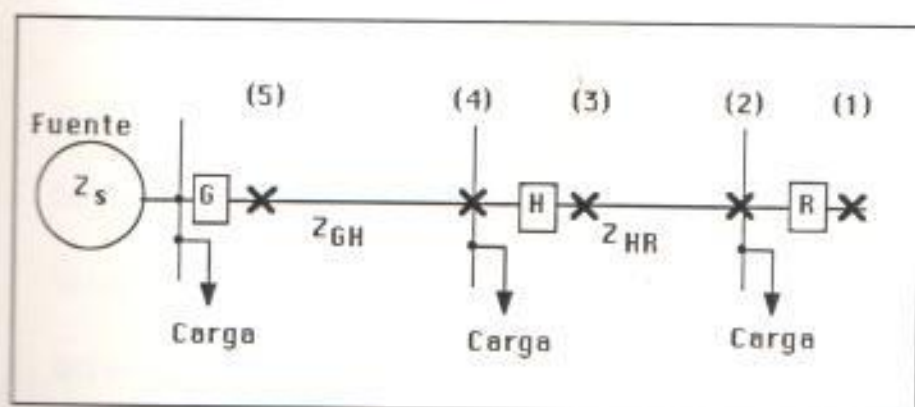


FIG. 1 Una Alimentadora Radial Típica

relevadores en **H** retarda la operación para fallas en (1) y (2). Si la falla es en (1), este retardo permite a los relevadores y disyuntores **R** operar antes que **H**. De esta manera, **H** opera para una falla en (1) (a menos que los relevadores **R** y los disyuntores asociados fallen), este operaría para una falla en (2). Esta técnica, llamada **coordinación o selectividad**, está diseñada para combinar los tiempos mínimos de operación para las fallas al comienzo de la línea (close-in fault) con un suficiente retardo para fallas remotas. En la FIG. 1, por ejemplo, los relevadores y disyuntores en **R** deben coordinar con aquellos a la derecha (no mostrados); **H** debe coordinar con **R**, y **G** con **H**. **Los relevadores son coordinados en pares.** Si la característica de disparo del relevador **H** ha sido ya coordinada con cualquier dispositivo de protección que exista en **R** y más allá, el disyuntor en **G** tiene que ser coordinado con aquellos en **H**.

Para los tres puntos críticos de falla - (5), (3), y (2) - se requieren los siguientes datos :

- a. Falla en (5) : corrientes máximas y mínimas de falla.
- b. Falla en (3) : corriente máxima de falla, la cual determina la coordinación requerida entre los disyuntores G y H .
- c. Falla en (2) : corrientes mínimas de falla, la cual determina cuando deben operar los relevadores G para proporcionar protección de respaldo para fallas en la línea HR no despejadas por el disyuntor en H .

1.23 COORDINACION

Los relevadores en el interior de un sistema pueden ser coordinados utilizando gráficos o tablas, aunque los gráficos son más utilizados en sistemas radiales. Papeles semi-log (absisa log para la corriente y ordenada lineal para el tiempo) o log-log son utilizados para la coordinación gráfica. El **papel log-log** es preferido cuando varios y diferentes tipos de dispositivos, incluidos fusibles, son coordinados en un

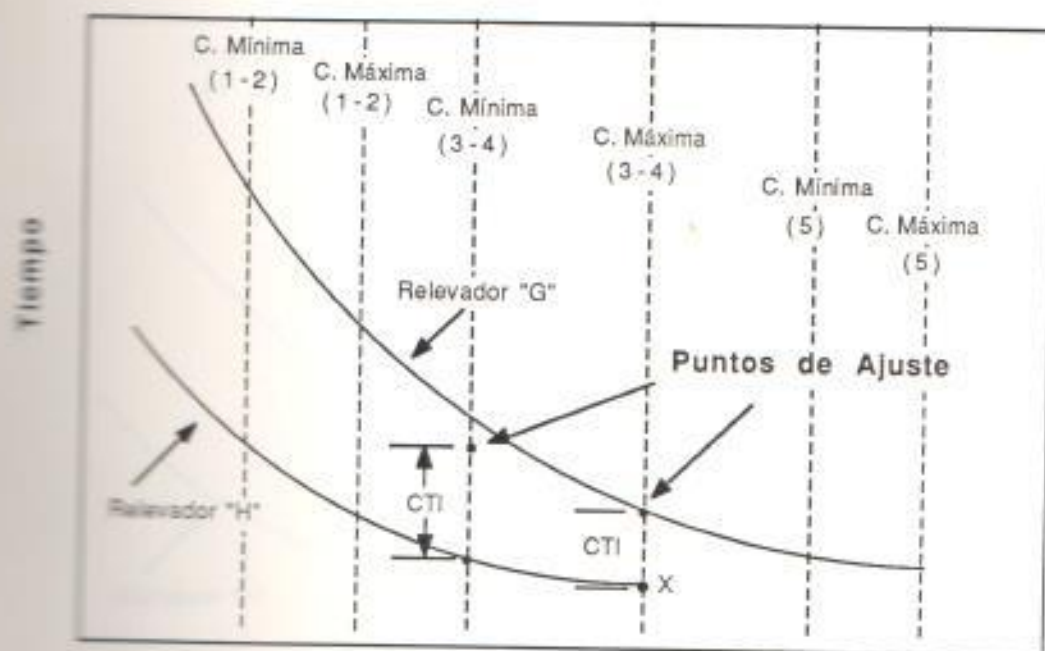
solo gráfico. La escala de la corriente puede ser en amperios primarios o en por unidad (p.u.). Cualquier diferencia en las relaciones de transformación de las corrientes debe ser tomada en consideración cuando se determine las corrientes actuales del relevador en diferentes posiciones.

El **procedimiento para la coordinación** se sigue como a continuación se describe (**FIG. 2A y FIG. 2B**).

Primero, se da por asumido que el tipo de relevador deseado (rango de tap y característica de tiempo) y la relación de transformación de la corriente han sido determinados. Entonces:

- a. Se determina las posiciones críticas de falla y los valores de corriente de falla.
- b. Se Grafica estas variables en la curva **tiempo-corriente** , dibujando líneas verticales en los diferentes valores.
- c. Se determina el tiempo de operación para el siguiente relevador hacia abajo (down-stream) para la máxima y mínima corrientes de falla. Para el

Corrientes de Falla



Corriente

Zona de Resp
para relevador "H"

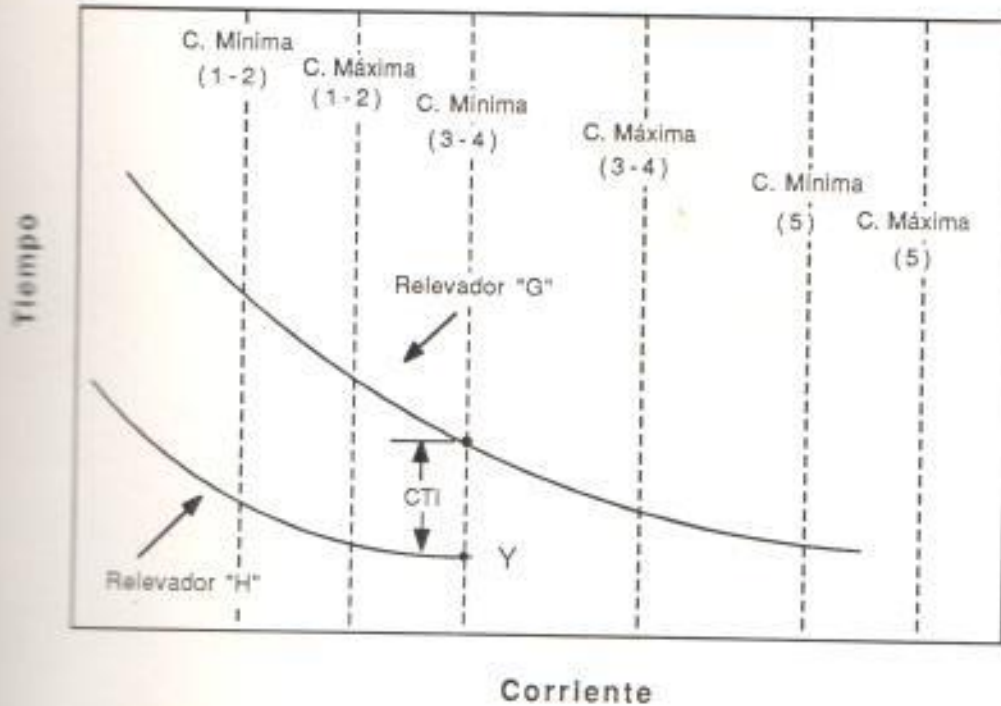
Zona de prot. Primaria
para relevador "H"

Zona de Prot. primaria
para relevador "G"

Zona de respaldo para el relevador "G"

FIG. 2.A Procedimiento de Ajuste de la Coordinación para los relevadores en el disyuntor (breaker) "G" de la FIG. 1, para fallas máximas.

Corrientes de falla



Zona de protecc. relev. "H"	Zona de Prot. primaria para relevador "H"	Zona de Protec. prim. para relevador "G"
	Zona de protección para relevador "G"	

FIG. 2.B Procedimiento de Ajuste de la Coordinación para los relevadores en el disyuntor (breaker) "G" de la FIG. 1, para fallas mínimas

relevador G, este tiempo de operación sería el mismo que para el relevador H en falla (3) máxima y mínima (mostradas como puntos X y Y, respectivamente). Se asume que el rele H ha sido ajustado (set) previamente para coordinar con los relevadores R y hacia la derecha.

- d. Se Adiciona el intervalo de tiempo de coordinación (CTI) a los puntos X y Y. Este paso da como resultado dos puntos de ajuste para la curva característica del relevador en G.
- e. Se selecciona una toma de corriente o derivacion (TAP) para el relevador G para que opere para falla (2) mínima y, para un relevador de fase, que no opere a máxima carga.

La falla (2) mínima debería operar el relevador en por lo menos dos veces la corriente de arranque (PICKUP). Para relevadores de fase, el ajuste debe ser siempre por encima de la máxima carga.

- f. Se selecciona un nivel de tiempo de tal manera que la curva tiempo-corriente del relevador G pase a

través o por encima una o dos veces de los puntos de ajuste X y Y y proporcione el tiempo mínimo de operación para la máxima y mínima falla (5). Este paso puede requerir ajuste de la toma de corriente (tap) y del nivel de tiempo.

Usualmente, la corriente máxima de falla es la más crítica para la coordinación.

- g. Se Repiten los pasos anteriores para cada sección de tiempo hacia arriba (up-stream).

1.24 INTERVALO DE TIEMPO DE COORDINACION { CTI }

El intervalo de tiempo de coordinación (**CTI - Coordination Time Interval**) es el mínimo intervalo que permite al relevador remoto y disyuntor despejar una falla en su zona de operación. Los factores que influyen el CTI son los siguientes:

- a. El tiempo que toma el disyuntor en interrumpir la falla.
- b. El tiempo que toma el impulso (impulse time-overtravel) del disco de inducción del

relevador después de que la corriente de falla ha sido interrumpida.

- c. El margen de seguridad para compensar las posibles desviaciones en las corrientes de falla calculadas, tiempo de operación del relevador, y errores en la relación de transformación de la corriente. Para coordinación por encima de aproximadamente 3 veces la corriente mínima de disparo, el CTI debería estar en el rango de 0.2 a 0.4 segundos. Los CTI mayores podrían ser utilizados en la parte alta de la curva para compensar los errores abajo de un multiplicador de tres. **Un CTI de 0.35 a 0.5 segundos es comunmente utilizado.**

1.2.5 SELECCION DE LA TOMA DE CORRIENTE (TAP) DE UN RELEVADOR DE SOBRECORRIENTE

Como se indicó anteriormente, los relevadores de sobrecorriente de fase no deben operar a la corriente mínima de carga que puede ocurrir en la línea. Las situaciones donde pueden suceder sobrecargas

temporales, deben ser tomadas en cuenta en el valor usado para ajustar los relevadores de sobre corriente. Así que, es importante para los ingenieros en protecciones junto con los ingenieros en operaciones determinar la corriente máxima posible para cada circuito. Este valor máximo (el menor tiempo de duración de la máxima carga) (**STM - Short Time Maximum Load**) puede diferir del rango de la línea, y es el valor que debería ser usado para el ajuste de los relevadores.

La toma o tap (valor mínimo de la corriente de arranque de los relevadores de sobre corriente de fase) debe ser por lo menos 2 veces la carga máxima normal y generalmente no menos de 1.5 veces. Asumiendo que el STM es mayor que la carga máxima normal, la toma de corriente (tap) puede seleccionarse como el próximo TAP disponible mayor que 1.25 STM. Los transformadores de corriente se escogen para proporcionar corrientes secundarias bajas durante la máxima carga nominal. Para permitir la operación en

presencia de posibles desbalances en un circuito normalmente balanceado, una buena regla práctica es ajustar los relevadores de fallas a tierra a no más del 10 por ciento de la máxima corriente de carga.

1.26 SELECCION DE LA CURVA DE TIEMPO DEL RELEVADOR DE SOBRECORRIENTE

Con la unidad de disco de inducción , se usan cinco (5) diferentes características en los circuitos de distribución, por ejemplo de la Westinghouse son :

- | | |
|---------------------------|---------|
| a. Mínimo definitivo | CO - 6 |
| b. Moderadamente inversa | CO - 7 |
| c. Inversa | CO - 8 |
| d. Muy inversa | CO - 9 |
| e. Extremadamente inversa | CO - 11 |

Estas curvas características tiempo-corriente se

muestran en la **FIG. 3**. Los ajustes de nivel de tiempo son seleccionados de tal manera que todos los relevadores operen en 0.2 seg. a 20 veces el ajuste de la toma de corriente (tap) (**MULTIPLES OF PICKUP**).

La selección de una característica de tiempo de un relevador es una función de las fuentes de energía, las líneas, y de las cargas. Ya que estos factores varían a través del sistema, la característica ideal para una línea y para una condición de operación requiere concesiones de las otras condiciones y líneas asociadas.

Si es posible, se deberían utilizar las curvas de tiempo con la misma o aproximadamente la misma característica. Curvas idénticas o similares aplicadas en diferentes lugares en el sistema tienden a seguir juntas como las condiciones de operación cambien. Si se tienen que utilizar diferentes características de curvas de tiempo, se deben chequear cuidadosamente todas las posibles condiciones de operación para

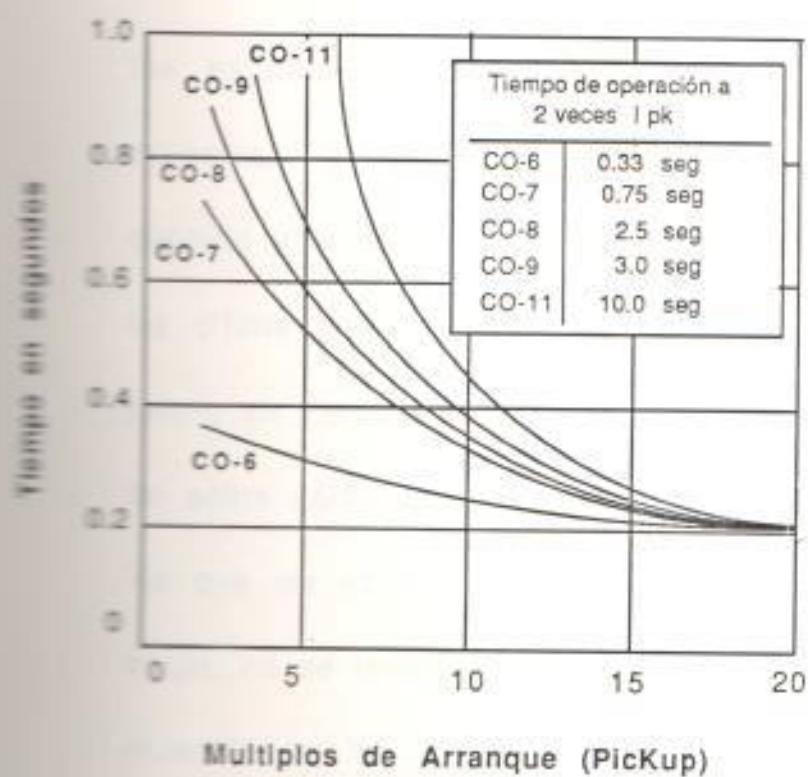


FIG. 3 Comparación de curvas de tipo CO

asegurar que el intervalo de tiempo de coordinación (CTI) se mantiene para un disparo selectivo. (Usar características similares, en otras palabras, minimiza los estudios de coordinación). En la FIG. 4 se muestran y comparan características de tiempo definido y tiempo inverso para un sistema ejemplo.

La última alimentadora que suministra energía a un centro de carga puede ser protegida con un dispositivo de sobre corriente instantáneo insertado en la carga.

Ya que no está envuelta ninguna coordinación en la carga, no se requiere de retardo en el tiempo -como se muestra en las gráficas tiempo-distancia- desde la barra R.

En la gráfica de la FIG. 4, la característica de tiempo definido se aproxima a la característica " **Mínimo definitivo**" (CO-6). El relevador en H está coordinado con el relevador R, y G con H, como se muestra en la misma figura.

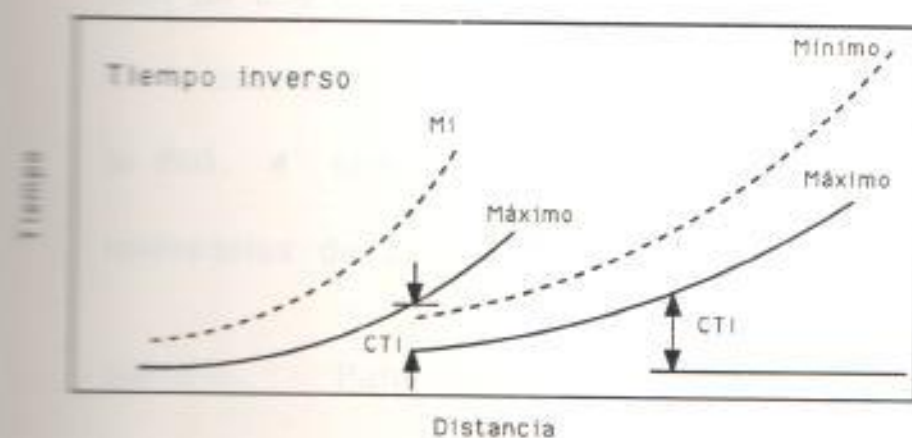
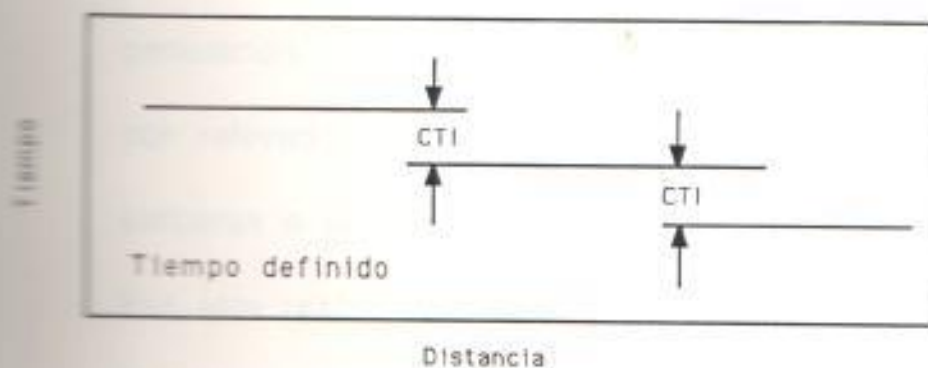
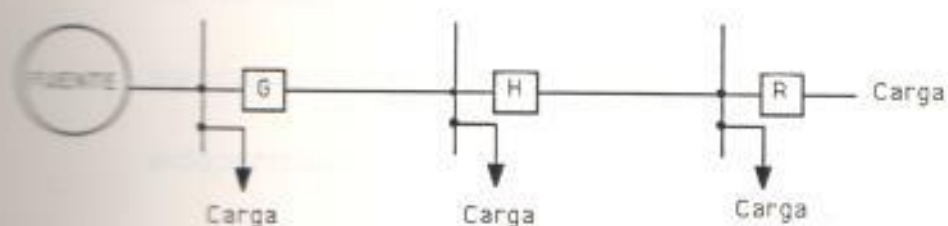


FIG. 4 Comparación entre relev. de sobrecorriente de Tiempo definido Vs. Tiempo Inverso para circuitos de alimentadoras radiales

La ventaja de este arreglo es que los tiempos de operación son relativamente constantes e independientes de los cambios en las magnitudes de las corrientes de falla desde máxima hasta mínima generación. Por otro lado, los tiempos de operación con relevadores de tiempo definido para grandes fallas cercanas a la fuente de generación son muy grandes. Por esta razón, este arreglo no es práctico donde hay más de una o dos alimentadoras radiales provenientes de la subestación de distribución. La gráfica inferior de la FIG. 4 nos muestra las características de relevadores de tiempo inverso aplicados al diagrama ejemplo. Para fallas cercanas al relevador, particularmente para condiciones de máxima carga, según como cambien las condiciones del sistema desde máxima a mínima carga, los tiempos de operación varían considerablemente. A pesar de que este arreglo puede producir tiempos de operación grandes para fallas mínimas cercanas a la barra remota.

La longitud de la línea es también un factor importante de considerar. Para una línea corta - en la cual su impedancia es pequeña comparada con la impedancia de la fuente - las corrientes de falla para fallas al comienzo de la línea (**close-in**) y al final de la línea (**far-end**) son esencialmente las mismas; esto es, la característica de tiempo inverso nos da un tiempo de operación relativamente fijo sobre la línea.

En estos casos, se prefiere la característica **mínimo definitiva**, ya que el tiempo de operación no varía mucho para diferentes niveles de generación como lo haría con relevadores de tiempo inverso.

En resumen, **las curvas (CO-6 Y CO-7) son más convenientes cuando:**

- a. No existen requerimientos de coordinación con los otros dispositivos de protección más lejanos en el sistema.
- b. La variación en las corrientes para fallas al comienzo y al final del circuito protegido es tan pequeña, que no se podría obtener beneficio de las

características de tiempo inverso.

- c. Las unidades de disparo instantáneo proporcionan buena cobertura.

Las curvas más inversas (CO-9 y CO-11) son más convenientes cuando :

- a. Las corrientes de falla son significativamente diferentes para fallas al comienzo de la línea (close-in) y al final de la línea (far-end). (Cuando la impedancia de la línea es grande comparada con la impedancia de la fuente, por ejemplo).
- b. Existe un apreciable flujo de corriente en la restauración del servicio (**cold load**).
- c. Cuando se requiere coordinar con otro tipo de dispositivos con características de tiempo inverso, tales como fusibles y reconectores (**reclosers**).

1.27 APLICACIONES DEL DISPARO INSTANTANEO

El añadir unidades de disparo instantáneo a relevadores de sobrecorriente proporciona operación de alta velocidad a los relevadores para fallas al comienzo de

la línea (close-in) y permitirá también ajustes más rápidos en los relevadores en la sección adyacente.

El disparo instantáneo debería ser utilizado en una alimentadora de distribución si la falla al comienzo de la línea (close-in fault) (máxima y/o mínima falla) está en el orden de 1.1 a 1.3 o más veces la máxima falla al final de la línea (far-end fault). En otras palabras, la unidad instantánea tiene que operar tantas veces como fallas en la línea pueda pero evitando la descordinación, no tiene que operar para la falla al final de la línea (far-end fault). Mientras mayor sea la razón de la falla al comienzo de la línea (close-in fault) para la falla al final de la línea (far-end fault), la unidad instantánea protegerá una parte mayor de la línea de transmisión.

1.2.8 RELEVADORES DE FASE Y DE TIERRA

Un circuito se puede proteger mejor usando tres relevadores de fase y uno de tierra. Con este arreglo, se puede remover un relevador para pruebas y el circuito puede ser energizado. Sin embargo, si sólo se

usan dos relevadores de fase y uno de tierra, el circuito podría salir de servicio durante las pruebas y mantenimiento.

1.2.9 COORDINACION FUSIBLE-RELE

A causa de que los fusibles tienen curvas cuya característica de tiempo es mucho más inversa que la mayoría de las características de los discos de inducción de sobrecorriente, puede resultar difícil la coordinación entre estos relevadores y los fusibles (FIG. 5). En esta figura están dibujadas una curva de fusible y dos juegos de curvas de relevadores - uno para el relevador CO-11 extremadamente inverso y el otro para el relevador CO-9 muy inverso - en una escala lineal/tiempo y logarítmica/corriente. Si las curvas de los relevadores son graficadas sobre la base de aproximadamente el mismo tiempo de operación a grandes valores de corriente, estas cruzarán la curva del fusible como se muestra. El juego de curvas de la derecha de los relevadores proporcionan un buen margen

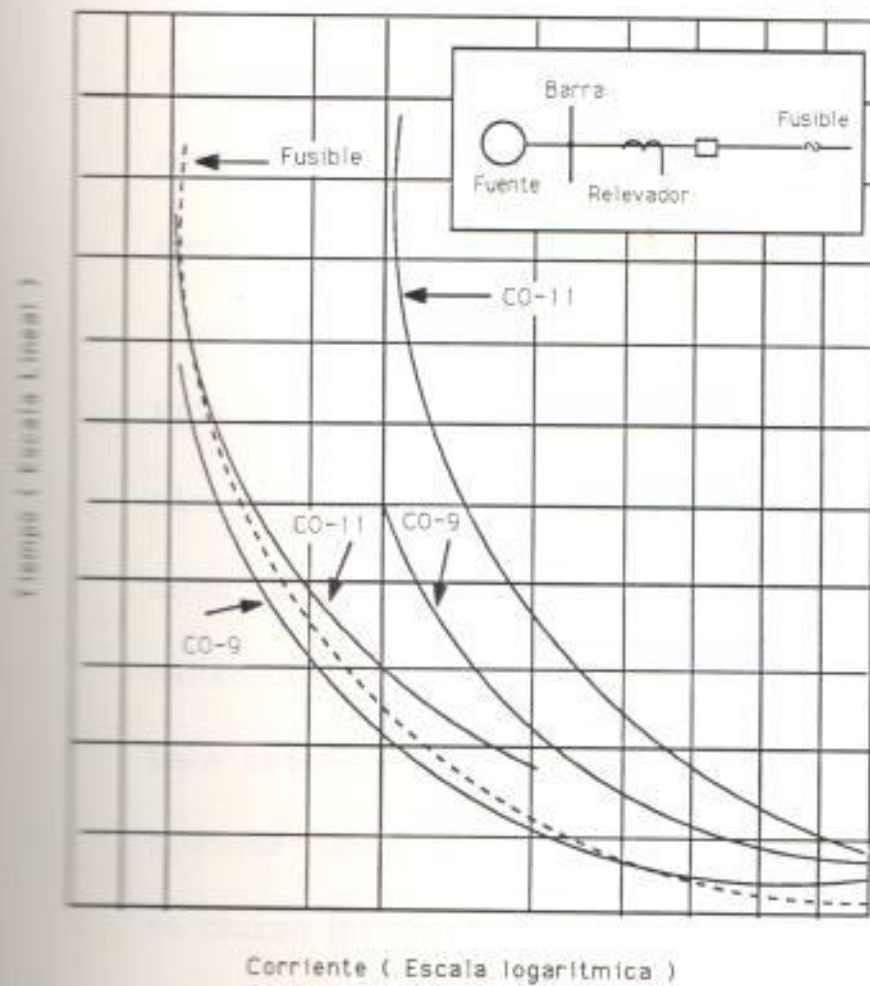


FIG. 5 Comparación entre las curvas de corriente del fusible y el relevador

de protección a altos valores de corriente de falla pero es insatisfactoriamente menor para valores medios de corriente de falla, particularmente con las características CO-11. Si la curva CO-11 de la derecha fuera movida hacia la izquierda, esta coordinaría mejor a valores menores de corriente. Como se muestra en la **FIG. 5**, la curva CO-11 cruzaría la curva del fusible para grandes valores de corriente de falla. Usualmente, tanto la característica muy inversa (CO-9) como la extremadamente inversa (CO-11) pueden ser ajustadas para coordinar con fusibles. Ajustando la toma de corriente (tap) y el nivel de tiempo de las áreas donde se cruzan las curvas como las mostradas en la **FIG. 5** impondrá condiciones de operación imprácticas e imposibles sobre el circuito.

Al graficar las curvas de fusibles, las sig. tres características de tiempo se deben considerar:

- a. El máximo tiempo que el fusible transporte corriente sin sufrir daño (fusión parcial), con el cual cambiarían sus características de operación

normal.

- b. El tiempo de fusión (**minimum melting time**) para las cintas fusibles (**fuse links**).
- c. El tiempo total que toma el fusible para despejar la falla (**maximum clearing time**).

Las dos primeras características son utilizadas para la coordinación con dispositivos de protección que se encuentren más allá del fusible. Normalmente, las curvas basadas en la característica de tiempo de fusión están equipadas con una banda de seguridad (**safety band**); en este caso, no se requieren curvas de tiempo máximo. La característica de tiempo de despeje (**clearing time**) es utilizada en su totalidad para coordinación con otros dispositivos de protección, incluidos relevadores, delante del fusible.

1.2.10 COORDINACION CON RECONECTADORES Y SECCIONALIZADORES

Los reconectores (**reclosers**) son dispositivos para la interrupción de circuitos, similares a los

disyuntores (**circuit breaker**), los que tienen disparo automático y facilidades de recierre. Normalmente, hay cuatro recierres antes de la apertura definitiva : uno instantáneo y tres ajustables, con retardo de tiempo; o lo que se conoce comúnmente como una rápida y tres lentas.

Hay tres tipos de control que son utilizados : disparos en serie (**series trip**), disparo por relevador (**relay trip**) y control estático (**static control**).

Los relevadores de disparo en serie y control estático tienen características de tiempo ajustable sobre un gran rango arranques (pickup) mínimos y áreas de curvas. Para simplificar la coordinación con relevadores de protección, los reconectores con disparo de relevador pueden ser equipados con cualquiera de los relevadores de tiempo de sobrecorriente con disco de inducción o instantáneos. Sin embargo, cuando sea posible, lo mejor es utilizar relevadores con el mismo tipo de característica de tiempo.

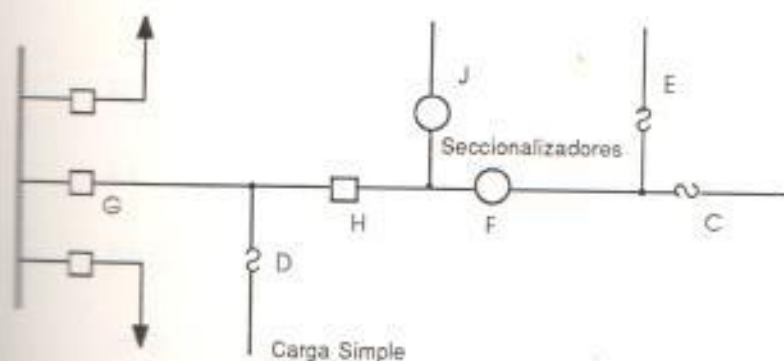
Los seccionadores son usualmente dispositivos de un polo, los cuales no tienen capacidad de interrupción de corriente de falla pero puede seccionar una alimentadora de distribución durante una falla permanente.

Los seccionadores simplifican la coordinación de los reconectores y fusibles, ya que, para corriente por sobre el mínimo disparo del reconector, el seccionador puede ser ajustado para abrirse para cualquier punto de corriente cero en el ciclo de reconexión. Esta secuencia asegura que el fusible no está sujeto a cualquier corriente de falla adicional.

1.2.11 SISTEMAS DE PROTECCION PARA ALIMENTADORAS DE DISTRIBUCION

La FIG. 6 muestra una alimentadora típica utilizando un disyuntor (breaker), reconector, y fusibles.

El disyuntor G deberá ser con retardo de tiempo para permitir el despeje de fallas más allá del reconector H. El primer despeje puede ser instantáneo, sin embargo, si las unidades de disparo



- Nota : G = Disyuntor
 H = Reconectador
 C, D y E = Fusibles
 J y F = Seccionalizadores

FIG. 6 Distribución típica de la protección de una alimentadora

instantáneo de los relevadores pueden ser ajustadas cerca de H. El reconectador en H puede ser ajustado para cualquier uno o dos recierres instantáneos; los otros dos o tres pueden tener retardo de tiempo, en total para 4 operaciones.

La unidad de disparo instantáneo opera el disyuntor G para fallas en el lado de fuente de D y H y resulta en una salida completa de la alimentadora. Para una falla permanente, en el lado de carga del fusible D el cual abrirá antes que G se dispare.

La FIG. 7 muestra las curvas de coordinación para los varios dispositivos de protección en la alimentadora de la FIG. 6. El reconectador H debe ser ajustado para operar para todas las fallas a lo largo de la alimentadora entre las ramas C y E y, por lo tanto, debe coordinar con los fusibles C y E. El seccionalizador F puede ser ajustado para abrirse durante el segundo o tercer período de apertura del reconectador H. Este arreglo permite que el servicio se restablezca al ramal J para fallas permanentes en

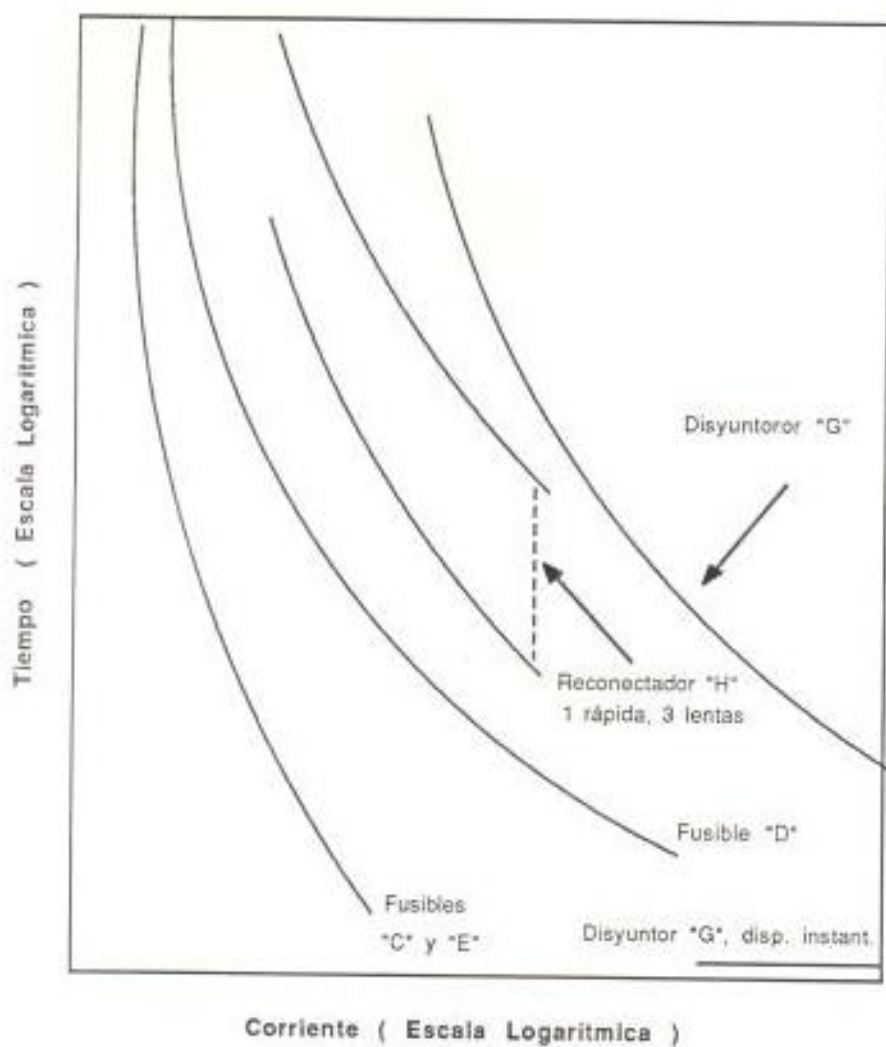


FIG. 7 Curvas de coordinación típicas para la protección de los dispositivos de la Alimentadora de la Fig. 6

el área entre F y C y F y E. Similarmente, el seccionalizador J debe ser también ajustado para abrirse durante el tercer período del reconectador H. En este sentido, el servicio puede ser restablecido para fallas permanentes en el ramal J.

La técnica de coordinación apropiada hace coincidir las características tiempo-corriente y mantiene un intervalo de tiempo, no menor que el CTI, entre los dispositivos sujetos a operación en la misma falla. Estas condiciones requieren un conocimiento de los valores de corrientes máximas y mínimas para varias fallas a lo largo de la alimentadora, así como también las características de los dispositivos de protección.

CAPITULO II

PROGRAMA DE COORDINACION UTILIZANDO RELEVADORES DE SOBRECORRIENTE

2.1 INTRODUCCION

El programa de coordinación es una herramienta de ayuda en la coordinación de los dispositivos de protección.

El programa denominado " **PROTEC** " permite tener acceso a la **Base de datos** de las curvas de los relevadores de sobrecorriente y habilitar estas curvas para ser dibujadas y manejadas en la **Hoja de Coordinación** y luego imprimir la coordinación entre cualquier par de dispositivos de protección.

La base de datos permite ser alimentada con un máximo de 100 archivos, c/u con 40 parejas de valores de puntos lo cual sirve para obtener las respectivas curvas y tal como se

encuentra al momento permite agregar un buen número de dispositivos.

La hoja de coordinación permite habilitar 5 curvas a la vez y manejarlas, desplazandolas en el eje de la corriente . La Hoja grafica en la pantalla una malla log-log donde son dibujadas las curvas y nos muestra un cuadro de comandos donde se encuentran los procedimientos de manejo, al igual que en la base de datos.

Y por último, el archivo Tutor nos explica acerca del uso del programa.

2.2 LENGUAJE DE PROGRAMACION UTILIZADO

El lenguaje de programación utilizado en el programa PROTEC es el **PASCAL**.

¿Porqué el Pascal en Sistemas de Potencia?

El Pascal tiene un buen tipo de datos, los cuales pueden ser creados e inventados, lo cual facilita la interacción entre el hombre y la máquina.

El Pascal tener una muy poderosa estructura de datos y

soporta la definición de tipos de datos por parte del usuario. También soporta listas enlazadas , arreglos, archivos, etc. Este lenguaje soporta muy bien la programación estructurada algo muy necesario en los actuales tiempos.

2.3 PRESENTACION Y MENU PRINCIPAL

El programa PROTEC debe ser cargado preferiblemente en un disco duro de una computadora IBM AT, la cual debe estar conectada a una impresora EPSON FX-1000.

Una vez invocado el programa nos mostrará a todo color la presentación inicial (" PROGRAMA PARA LA COORDINACION DE LA PROTECCION DE LINEAS DE DISTRIBUCION POR MEDIO DE RELEVADORES DE SOBRECORRIENTE ").

El Menú principal nos presenta cuatro opciones (FIG. 8) :

- A.- BASE DE DATOS
- B.- HOJA DE COORDINACION
- C.- TUTOR
- D.- SALIR AL D.O.S.

PROGRAMA PARA LA COORDINACION DE LA PROTECCION
DE LINEAS DE DISTRIBUCION UTILIZANDO
RELEVADORES DE SOBRECORRIENTE

MENU PRINCIPAL

A. BASE DE DATOS
B. HOJA DE COORDINACION
C. TUTOR
D. SALIR AL MENU PRINCIPAL

OPCION : _

EGBERTO RAMOS ROJAS

FIG. 8 VISTA DE LA PRESENTACION Y
MENU PRINCIPAL DEL PROGRAMA

2.4 BASE DE DATOS

2.4.1 MENU DE LA BASE DE DATOS Y GENERALIDADES

Una vez que se activa la opción A. del menu principal se entra a la base de datos del programa.

La base de datos permite almacenar hasta 100 archivos de parejas de datos distintos, las cuales se graban en disco al pasar a otra opción.

Este archivo permite almacenar hasta 40 parejas de puntos en dos pantallas de 20 parejas cada una (**FIG. 9**) ; las pantallas presentan un diseño especial y el enrejado **Xtap vs. t** no acepta otro tipo de caracteres que no sean números, esto es, cualquier tecla que el usuario presione, si no es número no será aceptada.

La Base de datos ofrece un menú de comandos muy apropiados para cualquier base de datos y nada complejos, como lo son:

- A. CREAR CURVA
- B. EDITAR CURVA
- C. ELEMENAR CURVA
- D. CATALOGO DE CURVAS
- E. FIN

2.4.2 OPCION CREAR CURVA

Esta opción una vez seleccionada, muestra la tabla en la pantalla xTAP vs. tiempo la cuál acepta los datos (**parejas de puntos**) de cada curva que se ingrese a la base de datos.

La opción ofrece funciones dadas por las teclas funcionales y otras más usadas en la mayoría de los programas desarrollados para computadoras.

Las funciones son muy útiles durante el almacenamiento de datos y estas son :

- F1** : GRABA DATOS LA CURVA EN LA MEMORIA RAM
- F4** : INSERTA UNA LINEA
- F5** : BORRA UNA LINEA
- F6** : ORDENA LOS DATOS DE MANERA ASCENDENTE
- PgUp** : VUELVE A LA PAG. ANTERIOR DE DATOS
- PgDn** : PASA A LA PAG. SIG. DE DATOS
- Home** : NOS LLEVA AL PRIMER CUADRO DE LA PAG.
- End** : NOS LLEVA AL ULTIMO CUADRO DE LA PAG.
- ↑** : ESTAS TECLAS NOS PERMITEN MOVERNOS VERTICALM.

↔ : ESTAS TECLAS NOS PERMITEN MOVERNOS HORIZONT.

Esc : CON ESTA TECLA CANCELAMOS TODA ACCION Y SALIMOS

La opción de crear curva solicita primero, el código con el cual se identifica posteriormente a la curva, este código consta de hasta 15 caracteres. Luego solicita la descripción, en la cual se explica el origen de la curva.

Así se empieza a ingresar los datos sólo numéricos, los datos de múltiplos de tap XTAP son sólo números de hasta dos cifras entera y dos decimales y los datos de tiempo son de hasta 3 enteros y un decimal.

Al ingresar un dato de XTAP se presiona retorno e inmediatamente se pasa a columna de tiempo.

Luego de esto se usan las teclas funcionales y se arreglan los datos ya ingresados para posteriormente ser grabados.

2.4.3 OPCION ELIMINAR CURVA

Esta opción permite eliminar una curva ya grabada en la base de datos, para lo cual solicita el código, y si este

no existe o está errado nos dará el correspondiente mensaje. Una vez identificada la curva da el código y la descripción correspondiente. Luego previo mensaje de confirmación la borra de la base de datos.

2.4.4 OPCION EDITAR CURVA

Esta opción permite llamar a los datos ya grabados en la base de datos para así poder hacer los cambios necesarios en las parejas de puntos.

Es importante anotar que una vez grabados los cambios en la curva, con la tecla funcional **F1**, el resto de registros que son iguales a cero, no son tomados en cuenta al momento de su graficación.

Las teclas funcionales resultan ser muy útiles dentro de esta opción, funciones como **INSERTAR LINEA**, permite ubicarse en cualquier línea de registro de pareja de puntos e introducir una línea en blanco en ese lugar, desplazando las otras un espacio hacia abajo.

La función **F6** permite **ORDENAR LOS DATOS** en forma

ascendente, algo muy útil, una vez que se ingresan nuevos datos a la curva.

La función ordenar es muy importante, pues muestra la posibilidad de haber cometido un error al introducir dos valores de XTAP iguales.

La función de BLANQUEAR REGISTRO permite, al ubicarse ya sea en un valor de XTAP o de tiempo, borrar automáticamente el contenido de este registro y así poder ingresar otro valor.

2.4.5 CATALOGO DE CURVAS

Al habilitar esta opción el programa, limpia la pantalla y presenta un archivo tipo texto del cual se obtendrá un listado completo de las curvas existentes en la base de datos, con sus códigos de identificación y descripciones.

Esta opción permite visualizar todas las curvas, y así no cometer el error de ingresar los datos otra vez.

Por otro lado, es el archivo de donde se habilitan las curvas a ser usadas en la Hoja de coordinación.

2.4.6 OPCION SALIR AL MENU PRINCIPAL (FIN)

Esta es la opción que permite salir de la base de datos al menú principal del programa.

En el cuadro de comandos, la opción se presenta con la letra E, la cuál sólo necesita ser digitada y de inmediato se sale de la base de datos.

2.5 HOJA DE COORDINACION

2.5.1 MENU DE HOJA DE COORDINACION Y GENERALIDADES

Al digitar la letra B del menú principal, se entra a la Hoja de coordinación.

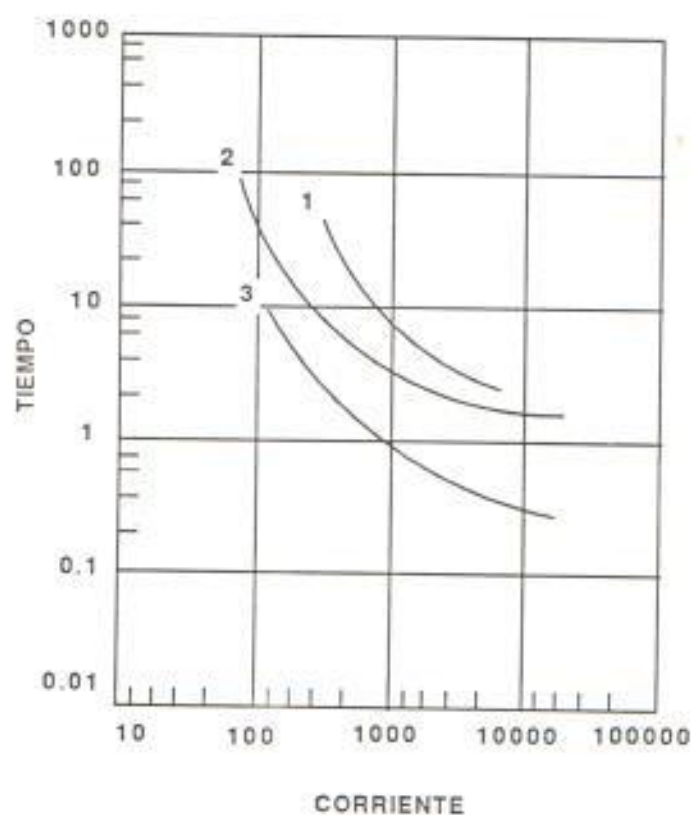
Hoja de coordinación es un procedimiento, que grafica una malla log-log, tanto en tiempo como en corriente. Sólo presenta la divisiones mayores.

El rango de la corriente en múltiplos de tap (eje X) va desde 0,1 hasta 100, y el de tiempo desde 0.01 seg.

hasta los 1.000 seg. , rangos lo suficientemente olgados para graficar cualquier comportamiento de dispositivos de protección de sobrecorriente.

La pantalla muestra un cuadro de comandos muy útiles en el manejo de la curvas, los cuales son (FIG. 10):

- A. HABILITA CURVAS
- B. LISTADO DE CURVAS
- C. DIBUJA CURVA
- D. ELIMINA CURVA
- E. DESPLAZA EN TIEMPO
- F. DESPLAZA EN CORRIENTE
- G. VALOR DE TIEMPO
- H. VALOR DE CORRIENTE
- I. COORDINACION
- J. IMPRIMIR HOJA
- K. SALIR AL MENU PRINCIPAL



HOJA DE COORDINACION	
A.	HABILITA CURVAS
B.	LISTA CURVAS
C.	DIBUJA CURVAS
D.	ELIMINA CURVA
E.	DESPLAZA EN TIEMPO
F.	DESPLAZA EN CORRIENTE
G.	VALOR DE TIEMPO
H.	VALOR DE CORRIENTE
I.	COORDINACION
J.	IMPRIMIR HOJA
K.	FIN
OPCION : <input type="text"/>	

CODIGO :

FIG. 10 MENU Y PRESENTACION DE LA HOJA DE COORDINACION

Dentro de las generalidades de la Hoja de coordinación está, que sólo se pueden habilitar hasta 5 curvas, para posteriormente ser graficadas.

La malla que presenta esta opción está hecha mediante el modo gráfico del Pascal y también habilitado el Comando Alta Resolución (HIRES).

El resto del capítulo tratará de las opciones y comandos de la Hoja de coordinación.

2.5.2 OPCION HABILITAR CURVAS

Este comando, al ser invocado, solicita los códigos de la curvas que se desea extraer de la base de datos, luego pregunta el TAP al cual va a trabajar el relevador y TD o retardo de tiempo, luego sigue preguntando por otras curvas que se desea graficar hasta darle la orden de no habilitar otra más. Al terminar de habilitar las curvas, estas están aptar para ser dibujadas y manejadas a gusto y conveniencia del usuario.

Para habilitar curvas sólo se debe digitar la opción A

del menú de comandos, y luego dejarse guiar por el programa, como en todas las presentaciones de este.

2.5.3 OPCION LISTADO DE CURVAS

Este comando muestra un listado de las curvas habilitadas (sus códigos) y con listado de las curvas graficadas.

Es una opción muy importante, por cuanto al ser graficadas las diferentes curvas, el programa incluye un número en la parte superior izquierda de la curva o más bien al inicio de la curva, con el cual será identificada por el programa y también por el usuario.

Con esta opción se reconoce las curvas graficadas en la malla log-log que se presenta en la pantalla del computador.

La opción muestra un pequeño recuadro en la parte superior derecha de la pantalla en la que aparecen primero las curvas habilitadas y luego las curvas graficadas con su identificación.

2.5.4 OPCION DIBUJA CURVAS

Este comando escoge de las curvas previamente habilitadas, las curvas que serán graficadas en la pantalla, para lo cual , al invocar el comando con la letra C, se debe dar el código de la curva a graficarse e inmediatamente se grafica la curva en la pantalla.

Si se digita un código incorrecto, o uno que no exista, el programa da mensaje de error, y se vuelve a repetir el paso anterior.

2.5.5 OPCION ELIMINA CURVAS

Es la opción o comando que permite borrar de la pantalla la curva que ya no se desee ver graficada.

Como en la opción anterior, al ser invocado el comando con la letra D, solicita el código de la curva que se "elimina" de la pantalla. Se digita el código y la curva es borrada de la pantalla al dar un Retorno.

El programa también protege la entrada de datos con mensajes de error.

2.5.6 OPCIONES DESPLAZAR EN TIEMPO Y DESPLAZAR EN CORRIENTE

Estas opciones permiten desplazar las curvas tanto en el eje de la corriente (X) como en el eje del tiempo (Y).

Al ser invocada cualquiera de las dos opciones del programa, con la letra E o F, el comando solicita como en casi todas las opciones, el código identificador de la curva, posteriormente lo hace con la constante de tiempo o corriente, según sea el caso, con la cual se desplaza la curva señalada. Luego el programa solicita el nuevo código con que es identificada temporalmente la nueva curva y procede a graficar la nueva curva, asignándole un número de identificación.

Esta nueva curva pasa al listado de curvas graficadas.

El comando de desplazamiento resulta ser muy importante y útil en el momento de coordinación.

2.5.7 OPCIONES DE VALOR DE TIEMPO Y VALOR DE CORRIENTE

Son opciones que permiten al usuario, interpolar valores en cualquier curva graficada.

La interpolación puede ser, para un valor de corriente dado, el programa entrega el valor de tiempo

correspondiente o para un valor de tiempo dado, el programa entrega el valor de corriente que le corresponde en la curva.

Estas opciones se habilitan con las letra G y H del menú de la hoja de coordinación.

El método de interpolación que se usa es el de punto y pendiente, de aquí la gran importancia de alimentar las curvas en la base de datos con tantos puntos como se pueda, en el programa la base de datos sugiere que sean 40 puntos.

Para esto se obtienen unos 15 puntos de las gráficas, a la vista, para luego utilizando un programa de regresión múltiple, como el SYSTAT que se encuentra grabado y habilitado en el CICYT.

Este programa da como resultado los coeficientes de la ecuación de la curva matemática, la cual puede ser evaluada para tantos puntos como se quiera, utilizando una calculadora programable y llenando el archivo de la base de datos con 40 puntos para así tener una muy buena resolución en la pantalla.

2.5.8 OPCION DE COORDINACION

Este es el comando, que una vez habilitado, permite realizar la coordinación entre pares de curvas de relevadores, para una corriente de falla dada.

La manera de ejecutar este procedimiento es como sigue :

Primero, se digita la letra I del menú de la hoja de coordinación. Luego el programa pregunta el código de la primera curva y luego el código de la segunda.

Pregunta después el valor de corriente para el cual se desea saber si las curvas coordinan o no.

La coordinación entre relevadores, no es otra cosa que, para una pareja de curvas dadas y un valor de corriente primaria de falla , evaluar las dos curvas las cuales deberán tener una diferencia o margen de tiempo no menor de 0.35 seg y no mayor de 0.5 seg.

Al terminar de ingresar los datos y dar un retorno el programa calcula inmediatamente el margen de tiempo

entre las curvas y da el mensaje de coordinación y el valor del margen de tiempo.

2.5.9 OPCION IMPRIMIR HOJA DE COORDINACION

Este es el comando que imprime las curvas y la hoja de coordinación con una muy buena resolución.

El comando al ser habilitado con la letra J, requiere información adicional del usuario como : La empresa eléctrica para la cual se realiza el estudio, el nombre de la alimentadora en la cual se encuentran los dispositivos de protección, la localización de esta alimentadora, el nivel de voltaje al cual trabaja dicha alimentadora, el dispositivo que se encuentra por encima y el dispositivo por debajo, el Ing. responsable de la coordinación, fecha y algún comentario que se considere necesario.

Al dar un retorno la computadora envía la señal a la impresora, la cual debe estar encendida, y esta procede a imprimir una malla log-log completa (divisiones mayores y menores) con las curvas que se coordinan y

su identificación, posteriormente en la parte inferior de la hoja imprime toda la información que solicitó anteriormente.

El programa esta listo para proseguir trabajando a las ordenes del usuario.

2.5.10 OPCION SALIR AL MENU PRINCIPAL

Es la opción que permite, una vez terminado el trabajo en la hoja de coordinación, salir al menú principal para así consultar el tutor o utilizar la base de datos o salir al D.O.S.

Para utilizar esta opción se debe digitar la letra K del menú de la hoja de coordinación.

2.6 ARCHIVO TUTOR

Es el archivo en el cual se encuentra escrito, toda la ayuda necesaria para trabajar en el programa.

Por otro lado el programa es muy fácil de utilizar y en el tutor se encuentra sólo la información absolutamente necesaria para el manejo del programa.

Para la lectura del tutor, es necesario encontrarse al nivel del menú principal y digitar la letra K.

Al encontrarse dentro del tutor, para su lectura, sólo es necesario presionar las teclas **PgUp** o para regresar sólo las teclas **PgDn**.

Una vez consultado el tutor, para salir debe presionarse la tecla **Esc** y con esto se regresa al menú principal, desde donde se escoge cualquier otra opción.

2.7 OPCION SALIR AL D.O.S.

Con esta opción una vez terminado todo el trabajo en el programa PROTEC , para regresar al nivel del sistema operativo de la computadora se debe encontrar a nivel del menú principal y digitar la letra D, inmediatamente el programa ejecutará el comando y saldrá del programa.

CONCLUSIONES Y RECOMENDACIONES

- 1.- El presente programa constituye una ayuda muy útil para los ingenieros eléctricos dedicados a las protecciones. Consiste en una herramienta general, que necesita ser complementada con el valioso aporte de otros estudiantes, los cuales pueden completar el programa con los módulos de coordinación entre relevadores y fusible, reconectores, etc.
- 2.- La creación de este programa demuestra que es conveniente la capacitación a los estudiantes en el campo de la programación, que sumada a sus conocimientos en sistemas de potencia, será una corriente productora de programas (software) los cuales resolverán muchos problemas en la vida profesional.
- 3.- Se recomienda que se incentive a los estudiantes para que en sus tesis de grado elaboren programas orientados al desarrollo de la Ing. Eléctrica.
- 4.- La presente versión del programa grafica y ayuda en la coordinación de relevadores de sobrecorriente, el siguiente paso está en completar el programa con las diferentes subrutinas para la coordinación entre otros pares de dispositivos de protección, como fusibles, reconectores, seccionalizadores, etc.

5.- Es necesario anotar que ni este ni otro programa es capaz de tener todas las respuestas para las innumerables contingencias que pueden ocurrir en un sistema de potencia.

BIBLIOGRAFIA

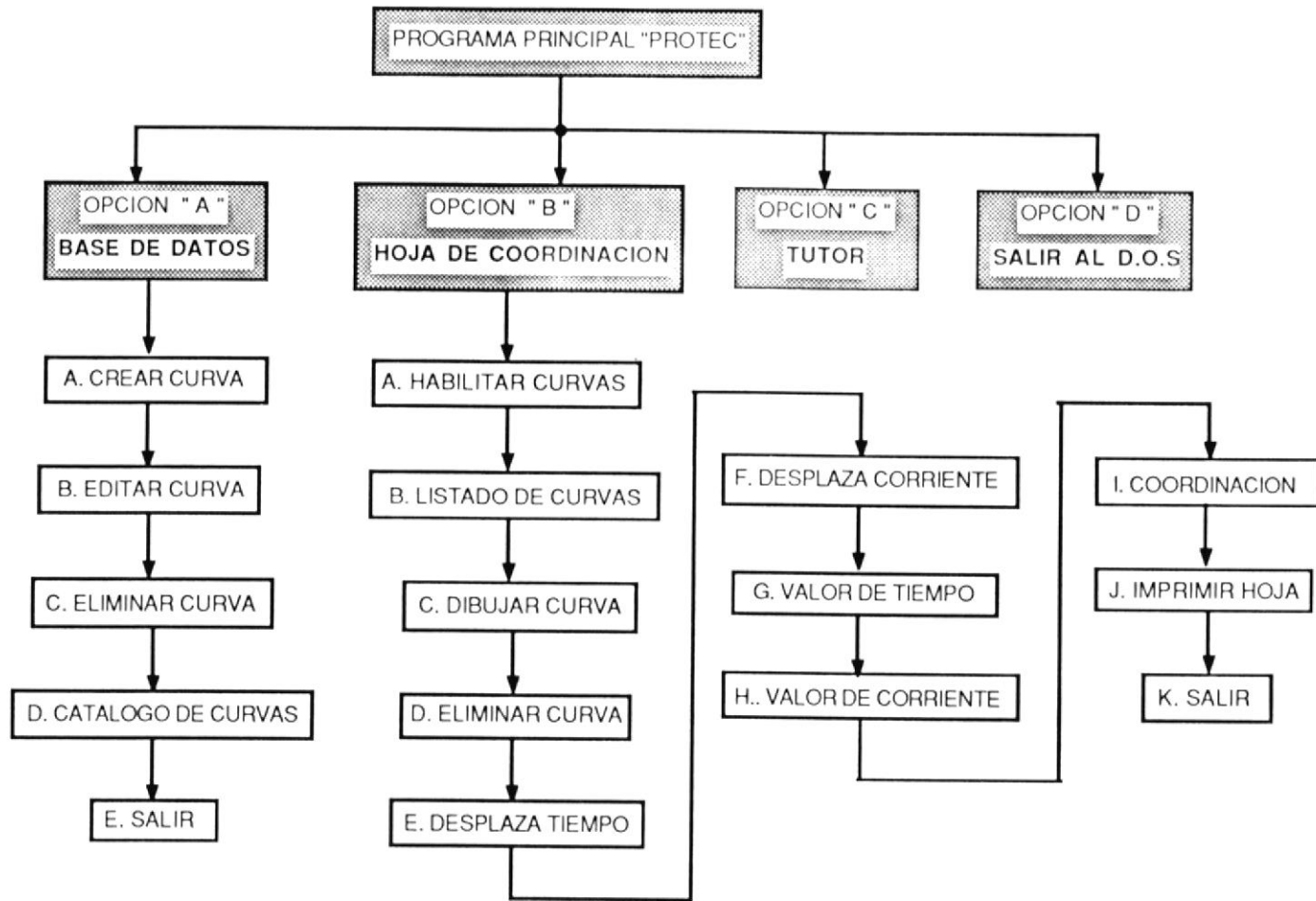
- 1.- GROGONO PETER, **PROGRAMACION EN PASCAL**, FONDO EDUCATIVO INTERAMERICANO, MEXICO D.F. , 1984
- 2.- ALBRECHT R. E., NISJA M. J., FERRO W. E., WAGNER C. L., ROCKEFELLER G. D., **DIGITAL COMPUTER PROTECTIVE DEVICE COORDINATION PROGRAM I - GENERAL PROGRAM DESCRIPCION**, IEEE TRANS ON PA&S, ABRIL 1964
- 3.- WILREKER V. F., WILSON R. A., **APPLIED PROTECTIVE RELAYING**, WESTINGHOUSE ELECT. CORP.
- 4.- **TURBO PASCAL, MANUAL DEL COMPILADOR**, ESPOL, 1986
- 5.- **SEMINARIO DE PROTECCIONES ELECTRICAS VOL. II**, FIE, ESPOL, 1985
- 6.- TORRES, CZITROM, **METODOS PARA LA SOLUCION DE PROBLEMAS CON COMPUTADOR DIGITAL. PROGRAMAS Y EJEMPLOS**, EDITORIAL REPRESENTACIONES Y SERVICIOS EN ING. S.A., MEXICO, 1980
- 7.- **V-PRO MCGRAW EDISON POWER SYSTEM DIVISION**

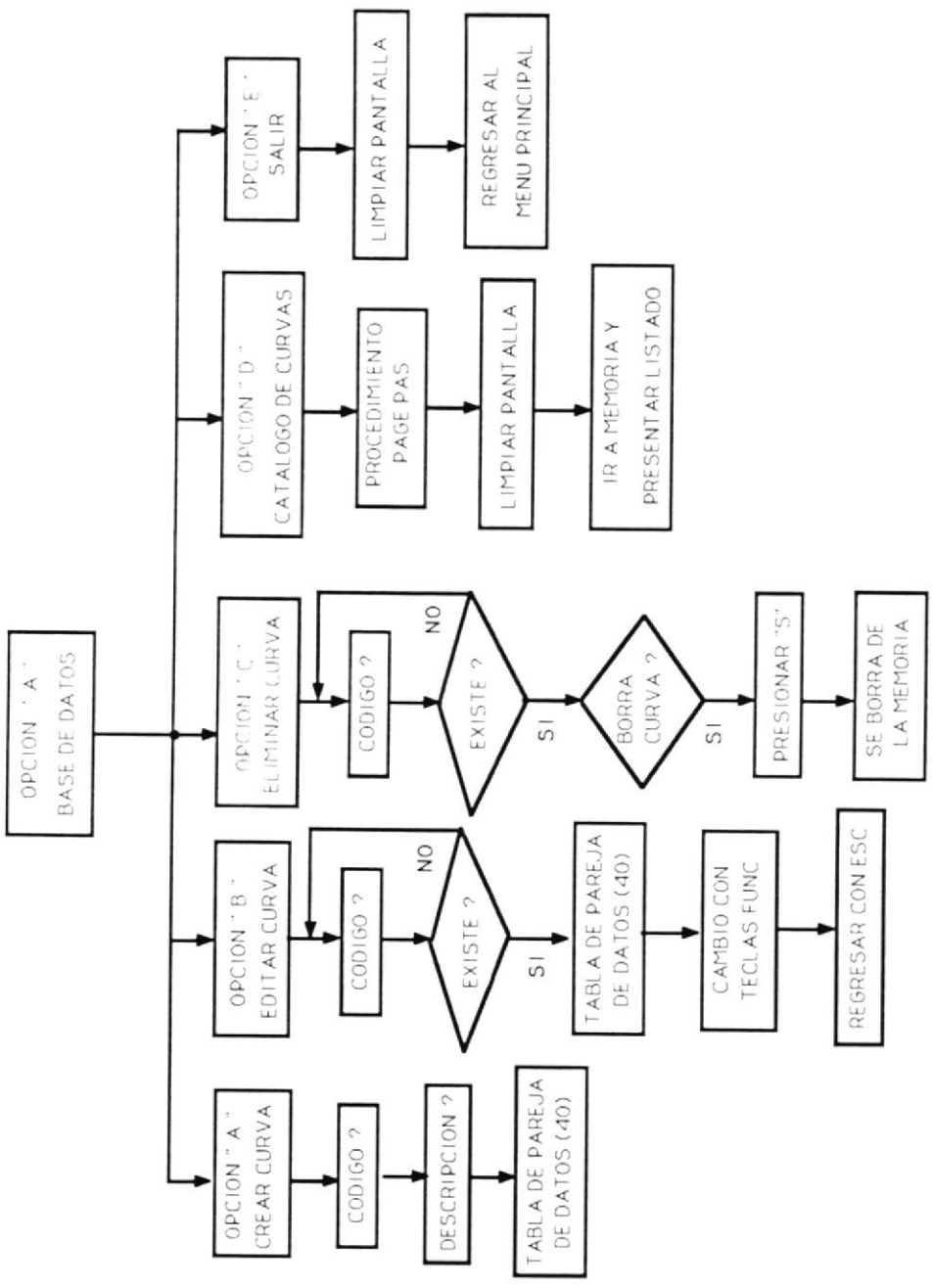
APENDICES

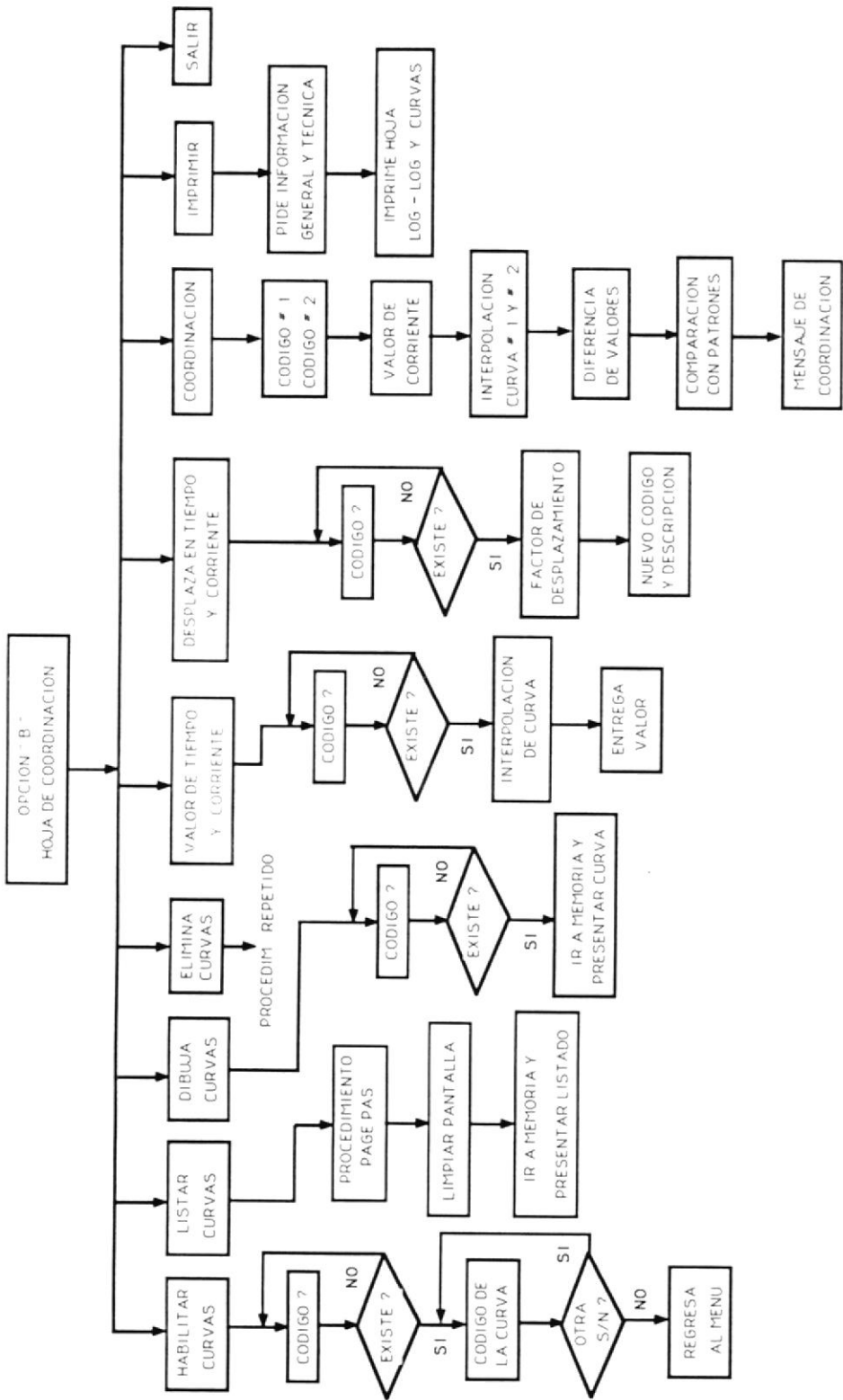
A.- DIAGRAMA D E BLOQUES DEL PROGRAMA

En el diagrama de bloques adjunto del las figuras se muestra cual es la secuencia lógica del programa.

En dicha figura se ven con claridad todas las opciones y comandos utilizados y disponibles en el programa PROTEC







B.- CORRIDAS EJEMPLO DEL PROGRAMA

Se toma como ejemplo el sistema de dos líneas radiales adyacentes de la FIG. 11.

En el sistema propuesto se asume que los relevadores de sobrecorriente son de tipo IAC (Curva CO8 de tipo inverso).

Análisis para el relevador en R

Ajuste del tap

$$I_{pk} = \frac{I_{f\text{mín}}}{1.5R_{TC}}, \text{ siempre y cuando la } I_{f\text{mín}} > I_{pc} \quad I \text{ plena carga} = [A]$$

$$I_{pk} = \frac{275}{1,5 (300/5)} = 3,06 \text{ A} \quad \text{Se escoge TAP} = 3$$

Ajuste de tiempo

$$X_{tap} = \frac{1500}{3x (300/5)} = 8,33$$

Asumiendo que el relevador debe operar a 0,8" , a fin de coordinar con el resto del Sistema se obtiene TD : 3

Este TD resulta de escoger de las curvas inversas de los relevadores, a un $X_{tap} = 8,3$ y tiempo de 0,8"

Se escoge la curva para el relevador en H, es una CO8 con un TD = 3, y TAP = 3 con una relación de

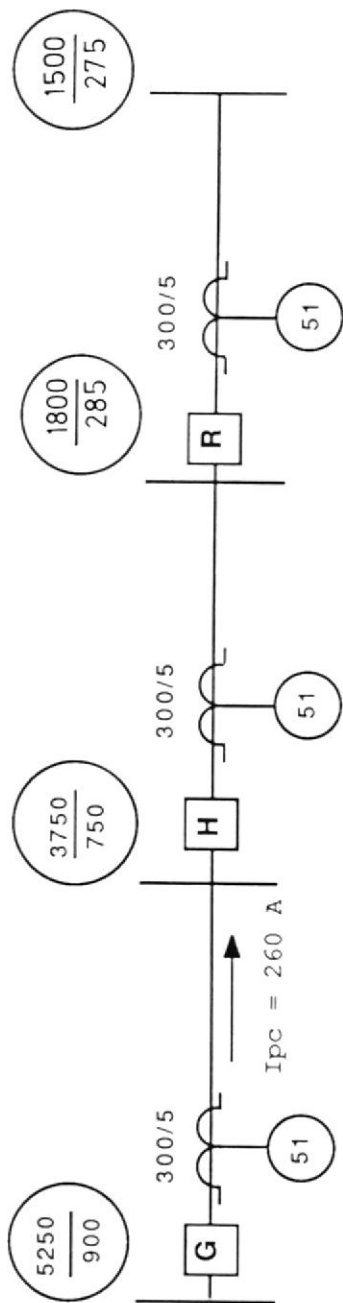


FIG. 11 EJEMPLO.- Protección de dos líneas Radiales Adyacentes

transformación de 300/5

Análisis para el relevador en H

Ajuste del tap

$$I_{pk} = \frac{I_{f\text{mín}}}{1.5R_{TC}}, \text{ siempre y cuando la } I_{f\text{mín}} > I_{pc}$$

$$I_{pk} = \frac{300}{1,5 (300/5)} = 3,33 \text{ A} \quad \text{Se escoge TAP} = 4$$

Ajuste de tiempo

$$X_{tap} = \frac{2500}{4 \times (300/5)} = 10,42$$

Se escoge la curva para el relevador en H, es una CO8 con un TD = 4, y TAP = 4 con una relación de transformación de 300/5

$$\text{Tiempo } t_R = 0,8 + 0,4 = 1,2$$

Análisis para el relevador en G

Ajuste del tap

$$I_{pu} = \frac{300}{300/5} = 5 \text{ A} \quad \text{Se escoge un TAP} = 5$$

Ajuste del tiempo

El punto de coordinación es para una falla en H, $t = 1,2''$

$$X_{tap} = \frac{3750}{5 \times (300/5)} = 12,5$$

$$t_R = 1,2 + 0,4 = 1,6''$$

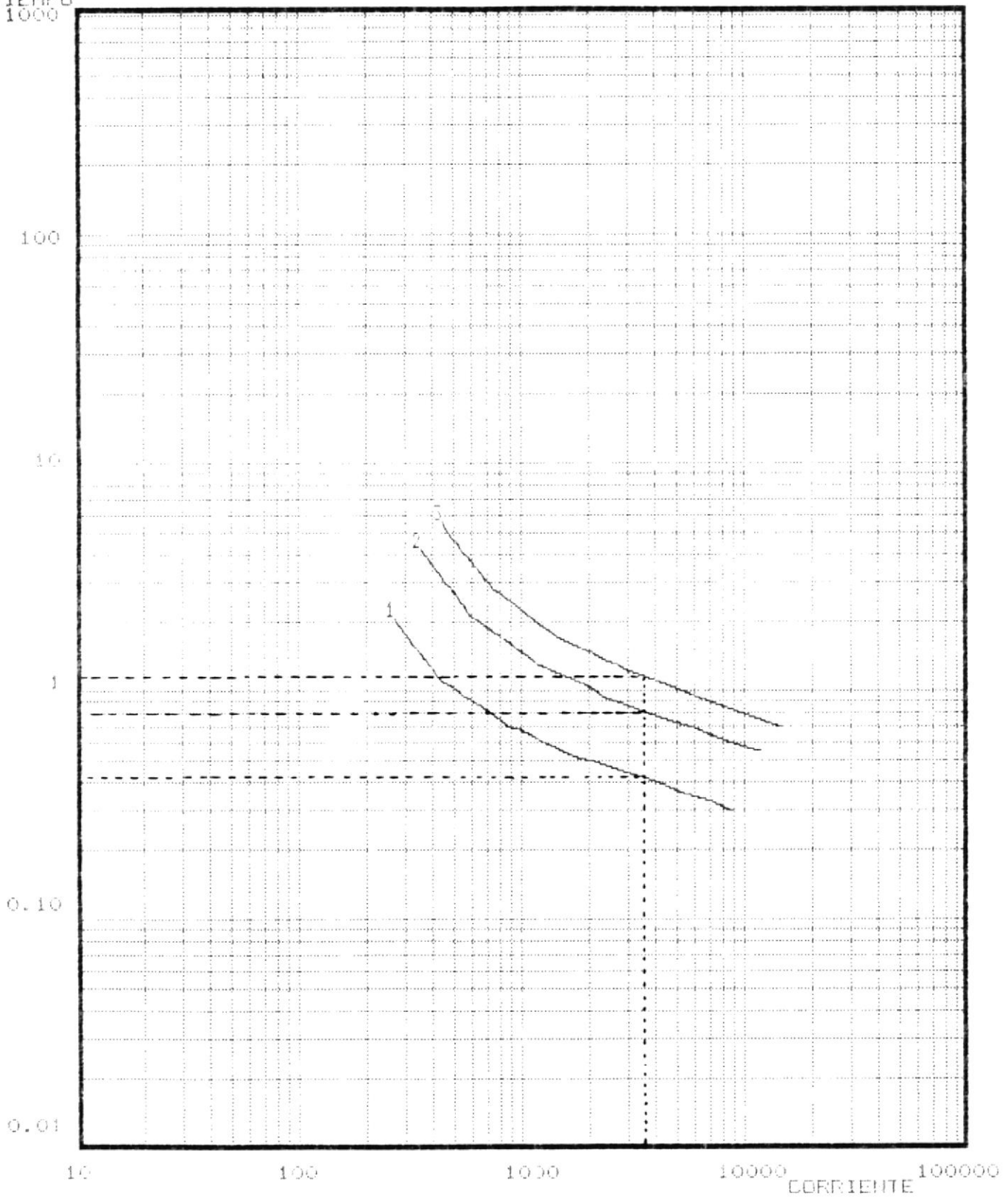
correspondiente a un TD = 5

Se escoge una curva de tipo inverso CO8 con un TD = 5, con un TAP = 5 y relación de transformación 300/5

Comprobar para corriente de falla 3.750 A.

Ver Corrida Hoja de Coordinación N° 1 del programa, adjunto, en la que se ven las curvas graficadas y la coordinación.

TIEMPO
1000



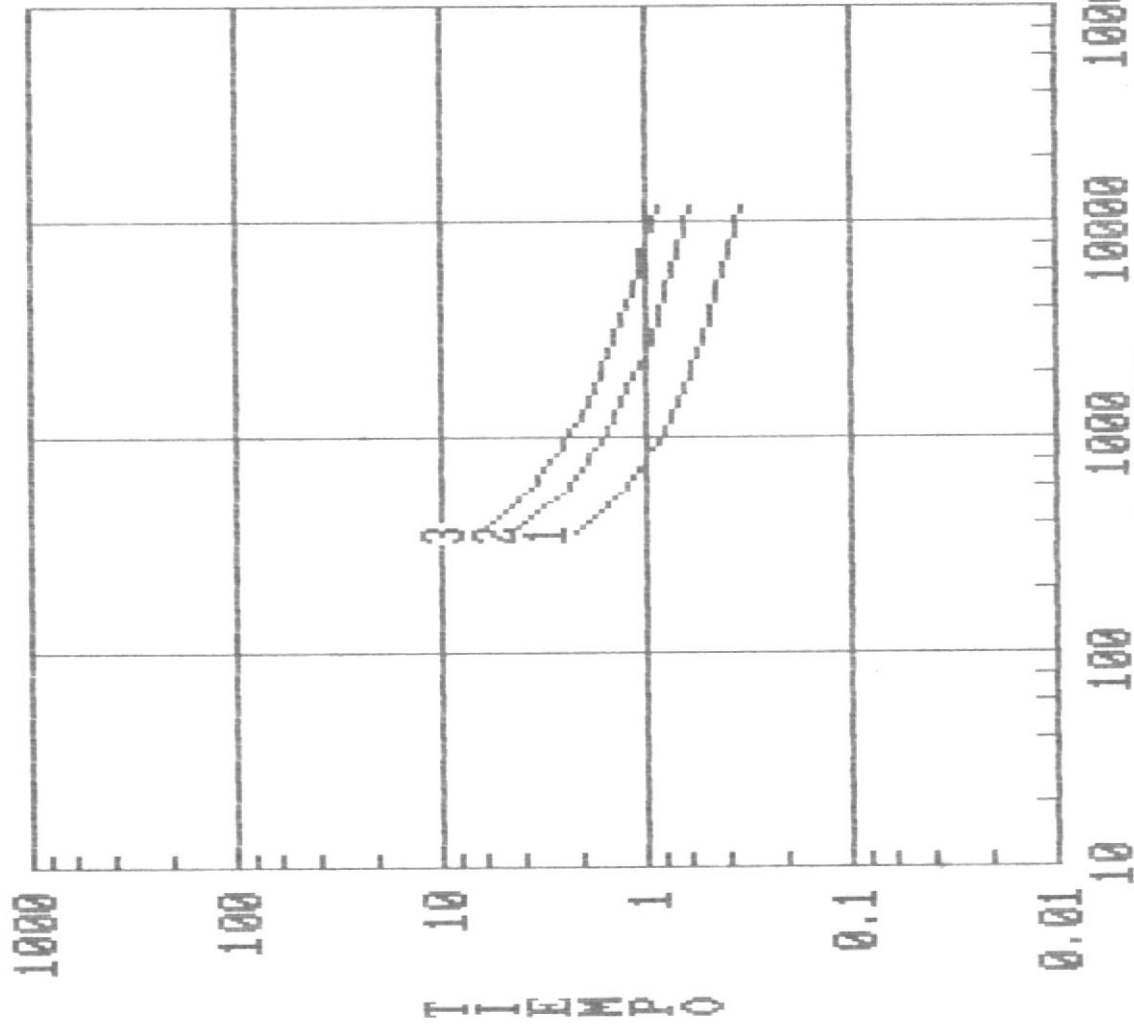
- 1 Código : C08-2
- 2 Código : C08-4
- 3 Código : C08-5

- Descripción : RELE INVERSO TD 2
- Descripción : RELE INVERSO TD 4
- Descripción : RELE INVERSO TD 5

Empresa : ESPOL
Alimentadora : PRUEBA
Dispositivo Upline :
Ing. Responsable :
Comentario :

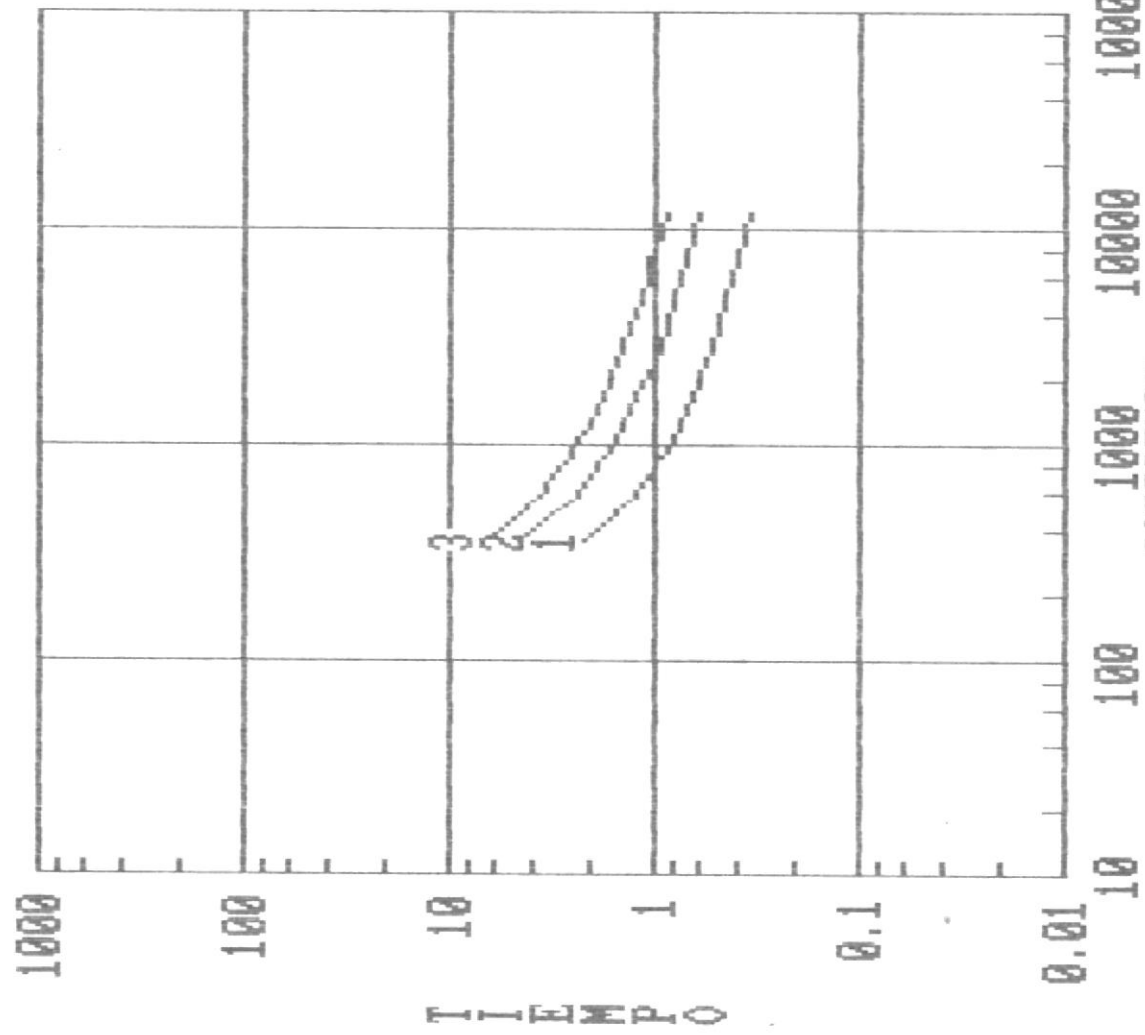
Departamento : F.I.E.
Localización : PRUEBA
Dispositivo Downline :
Fecha : 7/9/87

Voltaje : 13.8 K



CORRIENTE : 3750.00 AA ; 0.50, BB : 0.88 SI COORDINAN !!
 MARGEN DE TIEMPO : 0.38, <ENTER> para continuar.

HOJA DE COORDINACION	
A.	HABILITA CURVAS
B.	LISTA DE CURVAS
C.	DIBUJA CURVA
D.	ELIMINA CURVA
E.	DESPLAZA EN TIEMPO
F.	DESPLAZA EN CORRIENTE
G.	VALOR DE TIEMPO
H.	VALOR DE CORRIENTE
I.	COORDINACION
J.	IMPRIMIR HOJA
K.	FIN
OPCION : I	



CORRIENTE : 3750.00 BB : 0.88, DD : 1.32 SI COORDINAN !?
 MARGEN DE TIEMPO : 0.44, <ENTER> para continuar.

HOJA DE COORDINACION	
A.	HABILITA CURVAS
B.	LISTA DE CURVAS
C.	DIBUJA CURVA
D.	ELIMINA CURVA
E.	DESPLAZA EN TIEMPO
F.	DESPLAZA EN CORRIENTE
G.	VALOR DE TIEMPO
H.	VALOR DE CORRIENTE
I.	COORDINACION
J.	IMPRIMIR HOJA
K.	FIN
OPCION : I	

Problema N° 2

Ejemplo de protección de respaldo a sistemas de Distribución

La **FIG. 12** muestra un ejemplo de protección de respaldo dada por el relevador **L** a los relevadores **M** y **N** del sistema de distribución.

Las corrientes de fallas se han marcado para los puntos de interés.

Los relevadores **M** y **N** son del tipo CO-8 . El ajuste de estos relevadores dependen del sistema de distribución y ha sido determinado TAP 10 y TD = 2.

El relevador **L** es un relevador diferencial para protección de barra y transformadores. Para este relevador la corriente de operación es la suma de las corrientes **B** y **C** .

Ajuste del relevador L

El relevador **L** debe ser coordinado con los relevadores **M** y **N**, para ello las máximas y mínimas corrientes seleccionadas son las señaladas en la **FIG 12**.

I de puesta en trabajo

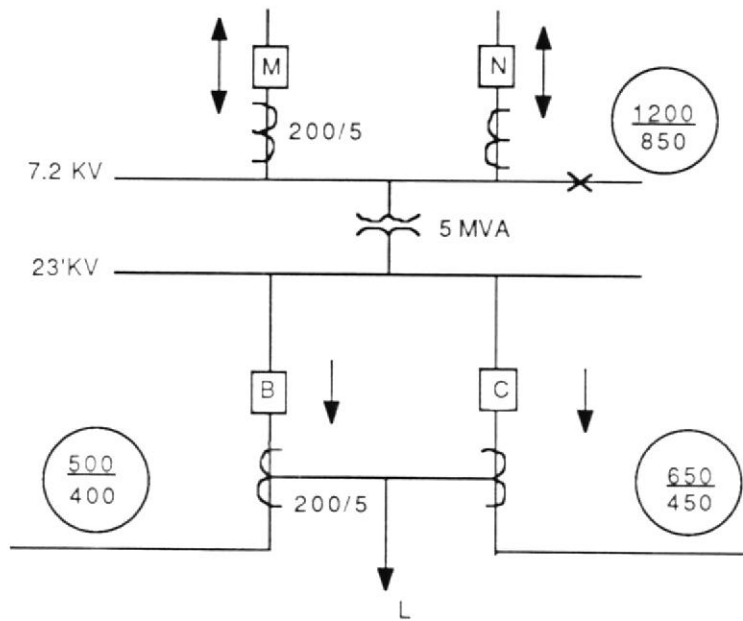


FIG. 12 Ejemplo 2.- Protección de respaldo para sist. de distribución

$$I_{pc} \text{ (TRANSF)} = \frac{5000}{23\sqrt{3}} = 125.5$$

$$I_{pc} \text{ (RELE)} = 125.5 \times 2 = 251 \text{ A}$$

$$I_{pk} = \frac{251}{400} = 6,28 \quad \textbf{TAP : 7}$$

Para 1200 A (máx. falla en 4)

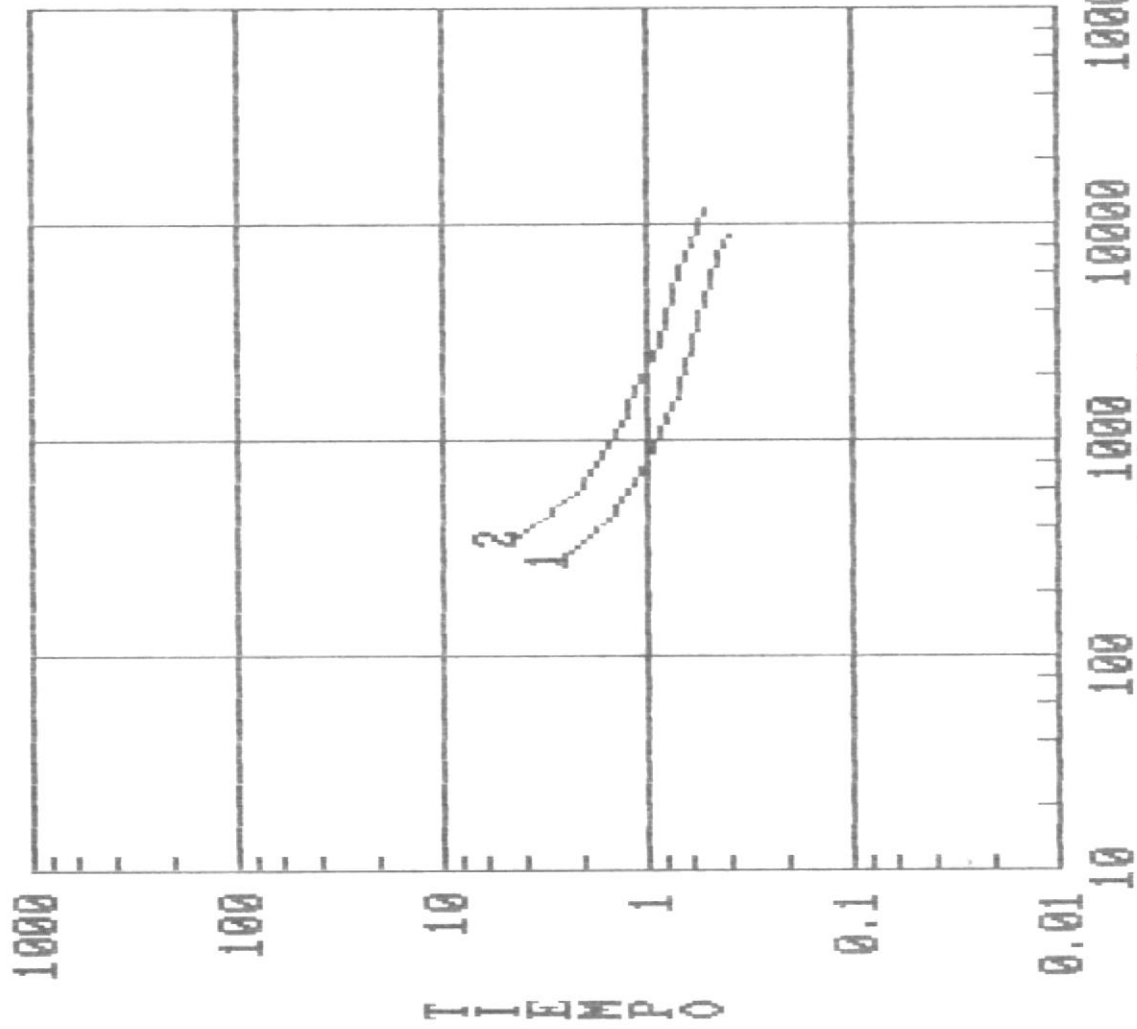
$$t_{R \text{ MN}} = 0,45 \text{ ''}$$

$$t_{R \text{ L}} = 0,45 + 0,4 = 0,85 \text{ ''}$$

$$X_{tap} : \frac{1200}{280} = 4,29 \quad \text{Corresponde a TD : 2}$$

Comprobar para corriente de falla 1.200 A.

Ver Corrida Hoja de Coordinación N° 2 del programa, adjunto, en la que se ven las curvas graficadas y la coordinación.



HOJA DE COORDINACION	
A.	HABILITA CURVAS
B.	LISTA DE CURVAS
C.	DIBUJA CURVA
D.	ELIMINA CURVA
E.	DESPLAZA EN TIEMPO
F.	DESPLAZA EN CORRIENTE
G.	VALOR DE TIEMPO
H.	VALOR DE CORRIENTE
I.	COORDINACION
J.	IMPRIMIR HOJA
K.	FIN
OPCION : I	

CORRIENTE : 1200.00 AA ; 0.85, B : 1.35 SI COORDINAN !
 MARGEN DE TIEMPO : 0.50, <ENTER> para continuar.

```

x,y,y1,i,j,k,inc_y: integer;
begin
  y:=0;
  inc_y := max_y*8 div 5;
  for i:= 1 to 5 do begin
    y := y+inc_y;
    for k := 1 to 9 do begin
      x := 0;
      y1 := y -round (inc_y*ln(k)/ln(10));
      for j:=1 to max_x div 3 do begin
        x := x+3;
        pset (x,y1);
      end;
    end
  end
end;

```

(=====)

```

procedure lineas_verticales;
var
  x,y,i,j,k,x1,inc_x: integer;
begin
  x := 0;
  inc_x := max_x div 4;
  for j:= 1 to 4 do begin
    x := x+inc_x;
    for k := 1 to 9 do begin
      y := 0;
      x1 := x -round (inc_x*(1-ln(k)/ln(10)));
      for j:=1 to max_y*4 do begin
        y := y+2;
        pset (x1,y);
      end
    end
  end
end;

```

(=====)

```

procedure m_neg(x1,y1,x2,y2: integer);
var
  i,j           : integer;
  dx,dy        : integer;
  acc_int,desp : integer;
  acc_real,paso : real;

```

```

begin
  dx := x2-x1;
  dy := y1-y2;
  x_current:=x1;
  y_current:=y1;
  paso :=dx/dy;
  acc_real :=0;
  acc_int :=0;

  (fija extremos)
  pset(x1,y1);
  pset(x2,y2);

  for l:= 1 to dy do begin
    acc_real:=acc_real+paso;
    desp :=trunc(acc_real)-acc_int;
    acc_int :=acc_int+desp;
    up(1);
    right(desp);
  end;
end;

(=====)
procedure m_pos(x1,y1,x2,y2: integer);
var
  i,j           : integer;
  dx,dy         : integer;
  acc_int,desp  : integer;
  acc_real,paso : real;

begin
  dx := x2-x1;
  dy := y2-y1;
  x_current:=x1;
  y_current:=y1;
  paso :=dx/dy;
  acc_real :=0;
  acc_int :=0;

  (fija extremos)
  pset(x1,y1);
  pset(x2,y2);

  for l:= 1 to dy do begin
    acc_real:=acc_real+paso;

```

```
    desp := trunc(acc_real)-acc_int;
    acc_int := acc_int+desp;
    down (1);
    right(desp);
end;
end;
```

```
(=====)
procedure vert(x,y1,y2: integer);
begin
    x_current := x;
    y_current := y1;
    pset(x,y1);
    pset(x,y2);
    down (y2-y1);
end;
```

```
(=====)
procedure horiz(x1,x2,y: integer);
begin
    x_current := x1;
    y_current := y;
    pset(x1,y);
    pset(x2,y);
    right (x2-x1);
end;
```

```
(=====)
procedure numero(x,y:integer;num : character);
var
    i,j : integer;
    pos : integer;
begin
    for i := 1 to 8 do begin
        pos := 128;
        for j := 1 to 8 do begin
            if (num [i]-pos) >= 0 then begin
                num [i] := num [i] - pos;
                pset (x-9+i,y-j);
            end
            else preset (x-9+i,y-j);
            pos := pos div 2;
        end;
    end;
end;
```



```

(=====)
procedure set_frame;
var
i,j: integer;
begin
  (limpia tabla)
  for i := 1 to max_x do
    for j := 1 to max_y do
      tabla[i,j] := 0;
      vert(1,1,max_y*8);           (traza linea lateral izquierda)
      vert(2,1,max_y*8);
      vert(3,1,max_y*8);
      vert(4,1,max_y*8);
      vert(max_x,1,max_y*8);       (traza linea lateral derecha)
      vert(max_x-1,1,max_y*8);
      vert(max_x-2,1,max_y*8);
      vert(max_x-3,1,max_y*8);
      horiz(1,max_x,1);           (traza linea superior)
      horiz(1,max_x,2);
      horiz(1,max_x,3);
      horiz(1,max_x,8*max_y);     (traza linea inferior)
      horiz(1,max_x,8*max_y-1);
      horiz(1,max_x,8*max_y-2);
end;

```

```

(=====)
procedure line (x1,y1,x2,y2:integer);
var
pendiente: real;
begin

```

```

  { caso 1: horizontal }
  if y2=y1 then begin
    if x1 < x2 then
      horiz(x1,x2,y1)
    else
      horiz(x2,x1,y1);
    end
  else begin
    pendiente :=(x2-x1)/(y2-y1);
    { caso 2: vertical }
    if pendiente=0 then begin
      if y1 < y2 then
        vert(x1,y1,y2)

```

```

for i := 2 to ultimo_dato do begin
  x2 := ln (datos [ i].corriente)/ln(10);
  y2 := ln (datos [ i].tiempo )/ln(10);
  jy2 := max_y*8-trunc((y2+2)/(5/(max_y*8)));
  ix2 := trunc((x2-1)/(4/max_x));
  line (ix1,jy1,ix2,jy2);
  ix1 := ix2;
  jy1 := jy2;
end;
end;
end;
end;

```

```

(=====)

```

```

begin
  uno [1] := 0; tres [1] := 0; cinco [1] := 0;
  uno [2] := 0; tres [2] := 65; cinco [2] := 199;
  uno [3] := 132; tres [3] := 129; cinco [3] := 133;
  uno [4] := 130; tres [4] := 129; cinco [4] := 133;
  uno [5] := 255; tres [5] := 137; cinco [5] := 133;
  uno [6] := 128; tres [6] := 149; cinco [6] := 133;
  uno [7] := 128; tres [7] := 99; cinco [7] := 121;
  uno [8] := 0; tres [8] := 0; cinco [8] := 0;
  dos [1] := 0; cuatro [1] := 0;
  dos [2] := 134; cuatro [2] := 16;
  dos [3] := 193; cuatro [3] := 24;
  dos [4] := 161; cuatro [4] := 20;
  dos [5] := 145; cuatro [5] := 18;
  dos [6] := 137; cuatro [6] := 255;
  dos [7] := 134; cuatro [7] := 16;
  dos [8] := 0; cuatro [8] := 0;
  textmode;
  clrscr;
  textcolor (lightred);
  write ('EMPRESA : ');
  textcolor (lightblue);
  readln (empresa);
  textcolor (lightred);
  write ('DEPARTAMENTO : ');
  textcolor (lightblue);
  readln (departamento);
  textcolor (lightred);
  write ('ALIMENTADORA : ');
  textcolor (lightblue);
  readln (alimentadora);

```

```

procedure coordinacion;

var
  codigo_curva1 : string[8];
  codigo_curva2 : string[8];
  pos1,pos2,i   : integer;
  tecla        : char;
  valor_tiempo1 : real;
  valor_tiempo2 : real;
  valor_corriente : real;
  diferencia    : real;
  cadena        : s255;
  posicion_error : integer;

begin
  gotoxy (1,24);
  write ('Codigo curva #1 : ');
  readln (codigo_curva1);
  pos1 := posicion_grafico(codigo_curva1);
  if pos1 > 0 then begin
    gotoxy (1,24);
    write (' ');
    gotoxy (1,24);
    write ('Codigo curva #2 : ');
    readln (codigo_curva2);
    pos2 := posicion_grafico(codigo_curva2);
    if pos2 > 0 then begin
      with curvas_grafico do begin
        gotoxy (1,24);
        write (' ');
        gotoxy (1,24);
        write ('Valor corriente : ');
        readln (cadena);
        val (cadena,valor_corriente,posicion_error);
        if posicion_error = 0 then begin
          valor_tiempo1 := interpola_tiempo (datos_curva[pos1],valor_corriente);
          valor_tiempo2 := interpola_tiempo (datos_curva[pos2],valor_corriente);
          if (valor_tiempo1>=0) and (valor_tiempo2>=0) then begin
            diferencia := abs (valor_tiempo1-valor_tiempo2);
            gotoxy (1,24);
            writeln (' ');
            gotoxy (1,24);
            if (diferencia >= 0.35 ) and (diferencia <= 0.5) then
              writeln (codigo_curva1,', ', 'codigo_curva2 ', ' SI COORDINAN !!!')
          end;
        end;
      end;
    end;
  end;
end;

```

```

else
    writeln (codigo_curva1, ', ', codigo_curva2, ' NO COORDINAN !!');
write ('MARGEN DE TIEMPO : ',diferencia:6:2,' ', <ENTER> para continuar.
repeat
    read (kbd,tecla);
until tecla = ^M;
end
else begin
gotoxy (1,25);
write ('Valor fuera de rango !!, <ENTER> para continuar. ');
repeat
    read (kbd,tecla);
until tecla = ^M;
end;
end;
else begin
gotoxy (1,25);
write ('Caracter invalido !!, <ENTER> para continuar. ');
repeat
    read (kbd,tecla);
until tecla = ^M;
end;
end;
end
else begin
gotoxy (1,25);
write ('No existe esa curva, <ENTER> para continuar. ');
repeat
    read (kbd,tecla);
until tecla = ^M;
end;
end
else begin
gotoxy (1,25);
write (' No existe esa curva, <ENTER> para continuar. ');
repeat
read (kbd,tecla);
until tecla = ^M;
end;
end;
end;

procedure habilita_curva;
const
numero_elementos =40;

```

```

numero_digitos_entero=6;
entero_ceros          = ' 0';

numero_digitos_decimal =6;
decimal_ceros         = ' 0.00';

type
  s6          =string(6);
  decimal    =string(numero_digitos_decimal);
  entero     =string(numero_digitos_entero );
  cod_curva=string(15);
  des_curva=string(30);
  msg       =string(40);          ((((((
  dato      =record
      codigo_curva      : cod_curva;
      descripcion_curva: des_curva;
      campo_01         : array[1..numero_elementos] of entero;
      campo_02         : array[1..numero_elementos] of decimal;
  end;

  datos_curvas=array[1..50] of dato;
  todos_los_datos = record
      ultima_curva: integer;
      datos       : datos_curvas;
  end;

var
  opcion      : char;
  i,j         : integer;
  file_var    : file of todos_los_datos;
  datos_disco : todos_los_datos;
  curvas_base : record
      ultima_curva : integer;
      datos_curva  : array [1..50] of linea;
  end;

function valor (st:s6):real;
var
  i,cod : integer;
  vlr   : real;
begin
  i:=1;
  while st[i] = ' ' do
    i := i+1;
  delete (st,i,i-1);

```



```

    val (st,vlr,cod);
    valor := vlr;
end;

function posicion_base(codigo : s8):integer;
var
    i,pos : integer;

begin
    i := 0;
    pos := -1;
    with curvas_base do begin
        while i <= ultima_curva do begin
            if datos_curva[i].codigo = codigo then pos := i;
            i := i+1;
        end;
    end;
    posicion_base := pos;
end;

procedure habilita_una;
var
    codigo_curva : string[8];
    pos          : integer;
    tecla       : char;

begin
    gotoxy (1,24);
    write (' ');
    gotoxy (1,24);
    write ('Codigo : ');
    readln (codigo_curva);
    pos := posicion_base(codigo_curva);
    if pos > 0 then begin
        if posicion_habilitadas (codigo_curva) < 0 then begin
            with curvas_habilitadas do begin
                if ultima_curva < 5 then begin
                    ultima_curva := ultima_curva+1;
                    datos_curva[ultima_curva] := curvas_base.datos_curva[pos];
                end
            end;
        else begin
            gotoxy (1,25);
            write ('Demasiadas curvas !!, <ENTER> para continuar. ');
            repeat
                read (kbd,tecla);
            until tecla = #13;
        end;
    end;
end;

```

```

        until tecla = ^M;
    end;
end;
end;
end
else begin
    gotoxy (1,25);
    write ('No existe ese codigo, <ENTER> para continuar. ');
    repeat
        read (kbd,tecla);
        until tecla = ^M;
    end;
end;
end;

begin
    assign (file_var,'base.dat');
    reset (file_var);
    read (file_var,datos_disco);
    close (file_var);
    with curvas_base do begin
        ultima_curva := datos_disco.ultima_curva;
        for i := 1 to ultima_curva do begin
            with datos_curva[i] do begin
                codigo      := datos_disco.datos[i].codigo_curva;
                descripcion := datos_disco.datos[i].descripcion_curva;
                j            := 1;
                while (datos_disco.datos[i].campo_01[j]<>entero_ceros) and
                    (datos_disco.datos[i].campo_02[j]<>decimal_ceros) do begin
                    datos[j].tiempo := valor (datos_disco.datos[i].campo_02[j]);
                    datos[j].corriente := valor (datos_disco.datos[i].campo_01[j]);
                    j := j+1;
                end;
                ultimo_dato := j-1;
            end;
        end;
    end;
    repeat
        habilita_una;
        gotoxy (1,25);
        write ('
        gotoxy (1,25);
        write ('Otra (S/N) ? ');
        repeat
            read (kbd,opcion);
            opcion := upcase (opcion);

```

```

        until opcion in ['S','N'];
        gotoxy (1,25);
        write ('
until opcion='N';
end;
(-----programa principal-----)
var
    fin : boolean;

begin
    curvas_habilitadas.ultima_curva := 0;
    curvas_grafico.ultima_curva    := 0;
    hires;
    imprime_menu;
    marco;
    fin := false;
    repeat
        case menu of
        'A' : habilita_curva;
        'B' : lista_curva;
        'C' : aumenta_curva;
        'D' : elimina_curva;
        'E' : desplaza_tiempo;
        'F' : desplaza_corriente;
        'G' : tiempo_valor;
        'H' : corriente_valor;
        'I' : coordinacion;
        'J' : imprime_hoja;
        'K' : fin := true;
        end;
        if not fin then dibuja_todas;
    until fin;
    textmode;
end;
);

```

```
(BASE
(mantiene la base de datos)
(27/jun/87 19:00)
```

```
procedure base;
```

```
const
esc      =27;           (codigos retornados por inkey.com)
pg_up   =195;
pg_dn   =203;
home    =193;
ed      =201;
arriba  =194;
abajo   =202;
izquierda=197;
derecha =199;
f1      =181;
f4      =184;
f5      =185;
f6      =186;
cero    =48;
nueve   =57;
punto   =46;
retro   =8;
enter   =13;
barra   =32;
A       =65;
Z       =90;
s       =115;
n       =110;
espace  = ';
```

```
lineas_por_pagina=20;
numero_paginas =2;
numero_elementos =40;
```

```
numero_digitos_entero=6;
entero_ceros      = ' 0';
entero_blanco     = ' ';
campo01_tope     = ' 0';
campo01_fondo    = '~~~~~'; (ascii: 254)
```

```
numero_digitos_parte_entera =3;
numero_digitos_parte_decimal=2;
```

```

numero_digitos_decimal      =6;
decimal_ceros                = ' 0.00';
decimal_blanco              = ' ';
campo02_tope                = ' 0.00';

numero_campos =2;
fila_referencia=3;
fila_pagina =25;
columna_pagina =5;

numero_registros=50;

type
decimal =string[numero_digitos_decimal];
entero   =string[numero_digitos_entero ];
cod_curva=string[15];
des_curva=string[30];
msg      =string[40];
dato     =record
    codigo_curva      : cod_curva;
    descripcion_curva: des_curva;
    campo_01          : array[1..numero_elementos] of entero;
    campo_02          : array[1..numero_elementos] of decimal;
end;
se       =string[6];

datos_curvas=array[1..numero_registros] of dato;
todos_los_datos= record
    ultima_curva: integer;
    datos       : datos_curvas;
end;

var
pagina,linea,fila,columna,tecla: integer;
com,campo,ref,cod,i,reg,j,k,p : integer;
columnas          : array[1..numero_campos] of integer;
registro,registro_ceros      : dato;
fin,conf,no_esta          : boolean;
codigo_curva              : cod_curva;
descripcion_curva         : des_curva;
datos                     : datos_curvas;
max_registro              : integer;
datos_disco                : todos_los_datos;
file_var                   : file of todos_los_datos;
file_txt                   : text;

```



```

        campo01:=registro.campo_01[j];
        registro.campo_01[j]:=registro.campo_01[k];
        registro.campo_01[k]:=campo01;
        campo02:=registro.campo_02[j];
        registro.campo_02[j]:=registro.campo_02[k];
        registro.campo_02[k]:=campo02;
    end;

    end;
end;

(recupera los 'ceros')
for j:=1 to numero_elementos do
begin
    if (registro.campo_01[j]=campo01_fondo) then registro.campo_01[j]:=campo01_tope;
    end;
end;(ordena_curva)

procedure despliegue_curva(var ref:integer);
var
k: integer;

begin
    (video normal)
    TextBackground(black);
    TextColor(LightGray);

    ref:=(pagina-1)*lineas_por_pagina;
    for k:=1 to lineas_por_pagina do
    begin
        GotoXY(columnas[1],fila_referencia+k);
        write(registro.campo_01[ref+k]);
        GotoXY(columnas[2],fila_referencia+k);
        write(registro.campo_02[ref+k]);
    end;

    GotoXY(columna_pagina,fila_pagina);
    write('Pagina ',pagina,'/',numero_paginas);
end;(despliegue_curva)

procedure tope;
begin
    linea:=1;
    campo:=1;
end;(tope)

```

```

procedure repeticion;
var
j,k : integer;
rtem: dato;

(elimina las abscisas repetidas, asume que los datos estan ordenados)
begin
  rtem:=registro_ceros;
  rtem.campo_01[1]:=registro.campo_01[1];
  rtem.campo_02[1]:=registro.campo_02[1];

  j:=1;
  for k:=2 to numero_elementos do
  begin
    if registro.campo_01[k]<registro.campo_01[k-1] then
      begin
        j:=j+1;
        rtem.campo_01[j]:=registro.campo_01[k];
        rtem.campo_02[j]:=registro.campo_02[k];
      end;
    end;

  registro:=rtem;
end;(repeticion)

procedure borra_linea;
var
k,p : integer;
rtem: dato;

begin
  rtem:=registro_ceros;
  p :=ref+linea;

  for k:=1 to p-1 do
  begin
    rtem.campo_01[k]:=registro.campo_01[k];
    rtem.campo_02[k]:=registro.campo_02[k];
  end;

  for k:=p+1 to numero_elementos do
  begin
    rtem.campo_01[k-1]:=registro.campo_01[k];
    rtem.campo_02[k-1]:=registro.campo_02[k];
  end;
end;

```

```
registro:=rtem;  
end;(borra_linea)
```

```
procedure inserta_linea;  
var  
k,p : integer;  
rtem: dato;
```

```
begin  
rtem:=registro_ceros;  
p :=ref+linea;  
  
for k:=1 to p-1 do  
begin  
rtem.campo_01[k]:=registro.campo_01[k];  
rtem.campo_02[k]:=registro.campo_02[k];  
end;  
  
for k:=p to numero_elementos-1 do  
begin  
rtem.campo_01[k+1]:=registro.campo_01[k];  
rtem.campo_02[k+1]:=registro.campo_02[k];  
end;  
  
registro:=rtem;  
end;(inserta_linea)
```

```
procedure fija_registro_ceros;  
begin  
registro_ceros.campo_01[ 1]:=' 0';  
registro_ceros.campo_01[ 2]:=' 0';  
registro_ceros.campo_01[ 3]:=' 0';  
registro_ceros.campo_01[ 4]:=' 0';  
registro_ceros.campo_01[ 5]:=' 0';  
registro_ceros.campo_01[ 6]:=' 0';  
registro_ceros.campo_01[ 7]:=' 0';  
registro_ceros.campo_01[ 8]:=' 0';  
registro_ceros.campo_01[ 9]:=' 0';  
registro_ceros.campo_01[10]:=' 0';  
registro_ceros.campo_01[11]:=' 0';  
registro_ceros.campo_01[12]:=' 0';  
registro_ceros.campo_01[13]:=' 0';  
registro_ceros.campo_01[14]:=' 0';  
registro_ceros.campo_01[15]:=' 0';
```

registro_ceros.campo_01[16]:=' 0';
registro_ceros.campo_01[17]:=' 0';
registro_ceros.campo_01[18]:=' 0';
registro_ceros.campo_01[19]:=' 0';
registro_ceros.campo_01[20]:=' 0';
registro_ceros.campo_01[21]:=' 0';
registro_ceros.campo_01[22]:=' 0';
registro_ceros.campo_01[23]:=' 0';
registro_ceros.campo_01[24]:=' 0';
registro_ceros.campo_01[25]:=' 0';
registro_ceros.campo_01[26]:=' 0';
registro_ceros.campo_01[27]:=' 0';
registro_ceros.campo_01[28]:=' 0';
registro_ceros.campo_01[29]:=' 0';
registro_ceros.campo_01[30]:=' 0';
registro_ceros.campo_01[31]:=' 0';
registro_ceros.campo_01[32]:=' 0';
registro_ceros.campo_01[33]:=' 0';
registro_ceros.campo_01[34]:=' 0';
registro_ceros.campo_01[35]:=' 0';
registro_ceros.campo_01[36]:=' 0';
registro_ceros.campo_01[37]:=' 0';
registro_ceros.campo_01[38]:=' 0';
registro_ceros.campo_01[39]:=' 0';
registro_ceros.campo_01[40]:=' 0';

registro_ceros.campo_02[1]:=' 0.00';
registro_ceros.campo_02[2]:=' 0.00';
registro_ceros.campo_02[3]:=' 0.00';
registro_ceros.campo_02[4]:=' 0.00';
registro_ceros.campo_02[5]:=' 0.00';
registro_ceros.campo_02[6]:=' 0.00';
registro_ceros.campo_02[7]:=' 0.00';
registro_ceros.campo_02[8]:=' 0.00';
registro_ceros.campo_02[9]:=' 0.00';
registro_ceros.campo_02[10]:=' 0.00';
registro_ceros.campo_02[11]:=' 0.00';
registro_ceros.campo_02[12]:=' 0.00';
registro_ceros.campo_02[13]:=' 0.00';
registro_ceros.campo_02[14]:=' 0.00';
registro_ceros.campo_02[15]:=' 0.00';
registro_ceros.campo_02[16]:=' 0.00';
registro_ceros.campo_02[17]:=' 0.00';
registro_ceros.campo_02[18]:=' 0.00';
registro_ceros.campo_02[19]:=' 0.00';

```

registro_ceros.campo_02[20]:=' 0.00';
registro_ceros.campo_02[21]:=' 0.00';
registro_ceros.campo_02[22]:=' 0.00';
registro_ceros.campo_02[23]:=' 0.00';
registro_ceros.campo_02[24]:=' 0.00';
registro_ceros.campo_02[25]:=' 0.00';
registro_ceros.campo_02[26]:=' 0.00';
registro_ceros.campo_02[27]:=' 0.00';
registro_ceros.campo_02[28]:=' 0.00';
registro_ceros.campo_02[29]:=' 0.00';
registro_ceros.campo_02[30]:=' 0.00';
registro_ceros.campo_02[31]:=' 0.00';
registro_ceros.campo_02[32]:=' 0.00';
registro_ceros.campo_02[33]:=' 0.00';
registro_ceros.campo_02[34]:=' 0.00';
registro_ceros.campo_02[35]:=' 0.00';
registro_ceros.campo_02[36]:=' 0.00';
registro_ceros.campo_02[37]:=' 0.00';
registro_ceros.campo_02[38]:=' 0.00';
registro_ceros.campo_02[39]:=' 0.00';
registro_ceros.campo_02[40]:=' 0.00';
end;(blanqueo_registro)

```

```

function menu:char;
var
  opcion : char;
begin
  repeat
    gotoxy (68,10);
    read (kbd,opcion);
    opcion := upcase (opcion);
  until opcion in ['A'..'E'];
  menu := opcion;
  write (opcion);
end;(menu)

```

```

procedure imprime_menu;
begin
  TextColor(green);
  GotoXY(52, 1);write('I#####');
  GotoXY(52, 2);write(':      BASE DE DATOS      :');
  GotoXY(52, 3);write('#####');
  GotoXY(52, 4);write(':  A. CREA CURVA      :');
  GotoXY(52, 5);write(':  B. EDITA CURVA    :');
  GotoXY(52, 6);write(':  C. BORRA CURVA    :');

```



```

GotoXY(1,12);write('      3      : PgDn: PAGINA SIGUIENTE');
GotoXY(1,13);write('      3      : END: ULTIMA PAGINA  ');
GotoXY(1,14);write('      3      :');
GotoXY(1,15);write('      3      :');
GotoXY(1,16);write('      3      :');
GotoXY(1,17);write('      3      :');
GotoXY(1,18);write('      3      :');
GotoXY(1,19);write('      3      :');
GotoXY(1,20);write('      3      :');
GotoXY(1,21);write('      3      :');
GotoXY(1,22);write('      3      :');
GotoXY(1,23);write('      3      :');
GotoXY(1,24);write('#####');
end;(titulos)

```

```

(=====programa principal=====)
begin

```

```

(parametros)
columnas[1]:=3;
columnas[2]:=12;
fija_registro_ceros;

```

```

(carga de la base)
assign(file_var,'base.dat');
reset(file_var);
read(file_var,datos_disco);
max_registro:=datos_disco.ultima_curva;
datos :=datos_disco.datos;
close(file_var);

```

```

fin:=false;

```

```

01:

```

```

repeat

```

```

  ClrScr;
  imprime_menu;
  case menu of

```

```

    (-----crea curva-----)

```

```

    'A' :

```

```

begin;

```

```

  (lee codigo y descripcion de la curva, verifica repeticion)

```

```

  if max_registro=numero_registros then begin mensaje('La base de datos esta llena');goto 01; end;

```

```

TextColor(LightBlue);
repeat
  gotoxy (24,23);
  write ('Codigo : ');
  readln (codigo_curva);
until codigo_curva<>copy(espace,1,length(codigo_curva));
for i:=1 to max_registro do
begin
  if datos[i].codigo_curva=codigo_curva then begin mensaje('Ya existe este codigo'); goto 01; end;
end;
gotoxy (24,24);
write ('Descripcion: ');
readln (descripcion_curva);

(carga de datos)
registro:=registro_ceros;

(inicializa variables)
pagina:=1;
linea :=1;
campo :=1;

titulos;
despliegue_curva(ref);

repeat
  fila :=fila_referencia+linea;
  columna:=columnas[campo];

  case campo of
1: lee_entero (tecla,registro.campo_01[ref+linea]);
2: lee_decimal(tecla,registro.campo_02[ref+linea]);
  end;

  case tecla of
f1: begin ordena_curva;repeticion; end;
f4: begin inserta_linea;despliegue_curva(ref); end;
f5: begin borra_linea;despliegue_curva(ref); end;
f6: begin ordena_curva;repeticion;despliegue_curva(ref);tope; end;
pg_up: begin pagina:=pagina-1;despliegue_curva(ref);tope; end;
pg_dn: begin pagina:=pagina+1;despliegue_curva(ref);tope; end;
home: begin pagina:=1;despliegue_curva(ref);tope; end;
ed: begin pagina:=numero_paginas;despliegue_curva(ref);tope; end;
enter: begin if campo=numero_camos then begin campo:=1;linea:=linea+1; end
else campo:=campo+1;

```

```

        end;
    arriba: linea:=linea-1;
    abajo: linea:=linea+1;
izquierda: begin if campo=1 then begin campo:=numero_campos;linea:=linea-1; end
            else campo:=campo-1;
            end;
    esc: begin
        confirma('cancela',conf);
        if not conf then tecla:=0;
        end;
    end;
until tecla in {f1,esc};

(grabacion)
if tecla=f1 then
    begin
        registro.codigo_curva :=codigo_curva;
        registro.descripcion_curva:=descripcion_curva;
        max_registro:=max_registro+1;
        datos[max_registro]:=registro;
    end;
end;
(-----edita curva-----)
'B':
begin;
    {lee codigo y descripcion de la curva, verifica existencia}
    if max_registro=0 then begin mensaje('La base de datos esta vacia');goto 01; end;
    TextColor(LightBlue);
    repeat
        gotoxy (24,23);
        write ('Codigo : ');
        readln (codigo_curva);
    until codigo_curva<>copy(espace,1,length(codigo_curva));
    no_esta:=true;
    for i:=1 to max_registro do
        begin
            if datos[i].codigo_curva=codigo_curva then begin no_esta:=false;reg:=i; end;
        end;
    if no_esta then begin mensaje('No existe este codigo'); goto 01; end;
    gotoxy (24,24);
    write ('Descripcion: ',datos[reg].descripcion_curva);

    (carga de datos)
    registro:=datos[reg];

```

```

(inicializa variables)
pagina:=1;
linea :=1;
campo :=1;

titulos;
despliegue_curva(ref);

repeat
    fila :=fila_referencia+linea;
    columna:=columnas(campo);

    case campo of
1: lee_entero (tecla,registro.campo_01(ref+linea));
2: lee_decimal(tecla,registro.campo_02(ref+linea));
    end;

    case tecla of
        f1: begin ordena_curva;repeticion; end;
        f4: begin inserta_linea;despliegue_curva(ref); end;
        f5: begin borra_linea;despliegue_curva(ref); end;
        f6: begin ordena_curva;repeticion;despliegue_curva(ref);tope; end;
pg_up: begin pagina:=pagina-1;despliegue_curva(ref);tope; end;
pg_dn: begin pagina:=pagina+1;despliegue_curva(ref);tope; end;
home: begin pagina:=1;despliegue_curva(ref);tope; end;
ed: begin pagina:=numero_paginas;despliegue_curva(ref);tope; end;
enter: begin if campo=numero_campos then begin campo:=1;linea:=linea+1; end
            else campo:=campo+1;
        end;
        arriba: linea:=linea-1;
        abajo: linea:=linea+1;
izquierda: begin if campo=1 then begin campo:=numero_campos;linea:=linea-1; end
            else campo:=campo-1;
        end;
        esc: begin
            confirma('cancela',conf);
            if not conf then tecla:=0;
        end;
    end;
until tecla in{f1,esc};

(grabacion)
if tecla=f1 then
    begin

```

```

        registro.codigo_curva      :=codigo_curva;
        registro.descripcion_curva:=descripcion_curva;
        max_registro:=max_registro+1;
        datos[reg]:=registro;
    end;
end;

(-----borra curva-----)
    '0' :
begin
    (lee codigo y descripcion de la curva, verifica existencia)
    if max_registro=0 then begin mensaje('La base de datos esta vacia');goto 01; end;
    TextColor(LightBlue);
    repeat
        gotoxy (24,23);
        write ('Codigo : ');
        readln (codigo_curva);
    until codigo_curva<>copy(espacio,1,length(codigo_curva));
    no_esta:=true;
    for i:=1 to max_registro do
    begin
        if datos[i].codigo_curva=codigo_curva then begin no_esta:=false;reg:=i; end;
    end;
    if no_esta then begin mensaje('No existe este codigo'); goto 01; end;
    gotoxy (24,24);
    write ('Descripcion: ',datos[reg].descripcion_curva);

    (carga de datos)
    registro:=datos[reg];

    (inicializa variables)
    pagina:=1;
    linea :=1;
    campo :=1;

    titulos;
    despliegue_curva(ref);

    (eliminacion)
    confirma('Elimina la curva',conf);
    if conf then
    begin
        for k:=reg+1 to max_registro do
        begin
            datos[k-1]:=datos[k];

```

```

procedure page(file_name:s7);

const
lpp=15;          (lineas por pagina)
bin_row = '
esc      =27;
pg_up    =195;
pg_dn    =203;
home     =193;
ed       =201;

type
cadena=string[70];
page_pointer=^page;
page= record
  row01: cadena;
  row02: cadena;
  row03: cadena;
  row04: cadena;
  row05: cadena;
  row06: cadena;
  row07: cadena;
  row08: cadena;
  row09: cadena;
  row10: cadena;
  row11: cadena;
  row12: cadena;
  row13: cadena;
  row14: cadena;
  row15: cadena;
  pag  : string[3];
  prev : page_pointer;
  next : page_pointer;
end;

var
first,current,last: page_pointer;
j,k,n_pag,comm    : integer;
file_var          : text;
registro          : cadena;
row               : array[1..15] of cadena;

procedure nueva_pagina;
var

```



```

npag: string[3];

begin
  n_pag:=n_pag+1;
  str(n_pag,3,npag);

  new(current);
  with current^ do begin
    row01:=row[ 1];
    row02:=row[ 2];
    row03:=row[ 3];
    row04:=row[ 4];
    row05:=row[ 5];
    row06:=row[ 6];
    row07:=row[ 7];
    row08:=row[ 8];
    row09:=row[ 9];
    row10:=row[10];
    row11:=row[11];
    row12:=row[12];
    row13:=row[13];
    row14:=row[14];
    row15:=row[15];
    pag :=npag;
  end;(with)
  if first=nil then begin
    first:=current;
    current^.prev:=nil;
    current^.next:=nil;
    last:=current;
  end
  else begin
    last^.next:=current;
    current^.prev:=last;
    current^.next:=nil;
    last:=current;
  end;(if)
end;(nueva_pagina)

procedure show_page;
begin
  gotoxy(1,7);
  with current^ do begin
    writeln(row01);
    writeln(row02);

```

```

        writeln(row03);
        writeln(row04);
        writeln(row05);
        writeln(row06);
        writeln(row07);
        writeln(row08);
        writeln(row09);
        writeln(row10);
        writeln(row11);
        writeln(row12);
        writeln(row13);
        writeln(row14);
        writeln(row15);
        gotoxy(1,22);
        write('Pagina ',pag,'/',n_pag);
    end;(with)
end;(show_page)

```

```

procedure read_comm;
var
ok: boolean;
begin
    repeat begin
        repeat
            inkey(comm);
        until comm in[esc,pg_up,pg_dn,home,ed];
        ok:=true;
        if (comm=pg_up) and (current=first) then ok:=false;
        if (comm=pg_dn) and (current=last) then ok:=false;
        if (comm=home) and (current=first) then ok:=false;
        if (comm=ed) and (current=last) then ok:=false;
    end;(repeat)
    until ok
end;(read_com)

```

```

(programa principal-----)
begin
    (load file-----)
    assign(file_var,file_name);
    reset(file_var);
    first:=nil;
    n_pag:=0;
    k :=0;

    if eof(file_var) then begin close(file_var);halt; end;

```

```

while not eof(file_var) do begin
  readln(file_var, registro);
  k:=k+1;
  row[k]:=concat(registro,
  if k=1pp then begin
    nueva_pagina;
    k:=0;
  end; (if)
  end; (while)
close(file_var);
if k>>0 then begin (pagina incompleta)
  for j:=k+1 to 1pp do begin
    row[j]:=din_row;
  end; (for)
  nueva_pagina;
end; (if)
(show file-----)
current:=first;
while true do begin
  show_page;
  read_comm;
  case comm of
    esc: exit;
    pg_up: current:=current^prev;
    pg_dn: current:=current^next;
    home: current:=first;
    ed: current:=last;
  end; (case)
end; (while)
end; (page)

```

```

        until opcion in ['S', 'N'];
        gotoxy (1,25);
        write ('
until opcion='N';
end;
(-----programa principal-----)
var
    fin : boolean;

begin
    curvas_habilitadas.ultima_curva := 0;
    curvas_grafico.ultima_curva := 0;
    hires;
    imprime_menu;
    mardo;
    fin := false;
    repeat
        case menu of
            'A' : habilita_curva;
            'B' : lista_curva;
            'C' : aumenta_curva;
            'D' : elimina_curva;
            'E' : desplaza_tiempo;
            'F' : desplaza_corriente;
            'G' : tiempo_valor;
            'H' : corriente_valor;
            'I' : coordinacion;
            'J' : imprime_hoja;
            'K' : fin := true;
        end;
        if not fin then dibuja_todas;
    until fin;
    textmode;
end;

```

```
(BASE
(mantiene la base de datos)
(27/jun/87 19:00)
```

```
procedure base;
```

```
const
esc      =27;           (codigos retornados por inkey.com)
pg_up    =195;
pg_dn    =203;
home     =193;
ed       =201;
arriba   =194;
abajo    =202;
izquierda=197;
derecha  =199;
f1       =181;
f4       =184;
f5       =185;
f6       =186;
cero     =48;
nueve    =57;
punto    =46;
retro    =8;
enter    =13;
barra    =32;
A        =65;
Z        =90;
s        =115;
n        =110;
espace   ='
```

```
lineas_por_pagina=20;
numero_paginas   =2;
numero_elementos =40;
```

```
numero_digitos_entero=6;
entero_ceros          =' 0';
entero_blanco        ='  ';
campo01_tope         =' 0';
campo01_fondo        ='~~~~~'; (ascii: 254)
```

```
numero_digitos_parte_entera =3;
numero_digitos_parte_decimal=2;
```

```

numero_digitos_decimal      =6;
decimal_ceros                = ' 0.00';
decimal_blanco              = ' ';
campo02_tope                = ' 0.00';

numero_campos =2;
fila_referencia=3;
fila_pagina =25;
columna_pagina =5;

numero_registros=50;

type
decimal =string[numero_digitos_decimal];
entero   =string[numero_digitos_entero ];
cod_curva=string[15];
des_curva=string[30];
msg      =string[40];
dato     =record
        codigo_curva      : cod_curva;
        descripcion_curva: des_curva;
        campo_01          : array[1..numero_elementos] of entero;
        campo_02          : array[1..numero_elementos] of decimal;
end;
s6       =string[6];

datos_curvas=array[1..numero_registros] of dato;
todos_los_datos= record
        ultima_curva: integer;
        datos       : datos_curvas;
end;

var
pagina,linea,fila,columna,tecla: integer;
coma,campo,ref,cod,i,reg,j,k,p : integer;
columnas          : array[1..numero_campos] of integer;
registro,registro_ceros      : dato;
fin,conf,no_esta      : boolean;
codigo_curva           : cod_curva;
descripcion_curva      : des_curva;
datos                  : datos_curvas;
max_registro          : integer;
datos_disco            : todos_los_datos;
file_var              : file of todos_los_datos;
file_txt              : text;

```



```

        campo01:=registro.campo_01[j];
        registro.campo_01[j]:=registro.campo_01[k];
        registro.campo_01[k]:=campo01;
        campo02:=registro.campo_02[j];
        registro.campo_02[j]:=registro.campo_02[k];
        registro.campo_02[k]:=campo02;
    end;
end;

    end;
end;

(recupea los 'ceros')
for j:=1 to numero_elementos do
begin
    if (registro.campo_01[j]=campo01_fondo) then registro.campo_01[j]:=campo01_tope;
    end;
end;(ordena_curva)

procedure despliegue_curva(var ref:integer);
var
k: integer;

begin
    (video normal)
    TextBackground(black);
    TextColor(LightGray);

    ref:=(pagina-1)*lineas_por_pagina;
    for k:=1 to lineas_por_pagina do
    begin
        GotoXY(columnas[1],fila_referencia+k);
        write(registro.campo_01[ref+k]);
        GotoXY(columnas[2],fila_referencia+k);
        write(registro.campo_02[ref+k]);
    end;

    GotoXY(columna_pagina,fila_pagina);
    write('Pagina ',pagina,'/',numero_paginas);
end;(despliegue_curva)

procedure tope;
begin
    linea:=1;
    campo:=1;
end;(tope)

```

```

procedure repeticion;
var
j,k : integer;
rtem: dato;

(elimina las abscisas repetidas, asume que los datos estan ordenados)
begin
  rtem:=registro_ceros;
  rtem.campo_01[1]:=registro.campo_01[1];
  rtem.campo_02[1]:=registro.campo_02[1];

  j:=1;
  for k:=2 to numero_elementos do
  begin
    if registro.campo_01[k]<>registro.campo_01[k-1] then
      begin
        j:=j+1;
        rtem.campo_01[j]:=registro.campo_01[k];
        rtem.campo_02[j]:=registro.campo_02[k];
      end;
    end;

  registro:=rtem;
end;(repeticion)

procedure borra_linea;
var
k,p : integer;
rtem: dato;

begin
  rtem:=registro_ceros;
  p :=ref+linea;

  for k:=1 to p-1 do
  begin
    rtem.campo_01[k]:=registro.campo_01[k];
    rtem.campo_02[k]:=registro.campo_02[k];
  end;

  for k:=p+1 to numero_elementos do
  begin
    rtem.campo_01[k-1]:=registro.campo_01[k];
    rtem.campo_02[k-1]:=registro.campo_02[k];
  end;

```

```

    while not eof(file_var) do begin
    readln(file_var,registro);
    k:=k+1;
    row[k]:=copy(concat(registro,'
if k=lpp then begin
    nueva_pagina;
    k:=0;
end;(if)
end;(while)

close(file_var);

if k>0 then begin (pagina incompleta)
    for j:=k+1 to lpp do begin
row[j]:=bin_row;
    end;(for)
    nueva_pagina;
end;(if)

(show file-----)
current:=first;
while true do begin
show_page;
read_com;
case com of
    esc: exit;
    pg_up: current:=current^.prev;
    pg_dn: current:=current^.next;
    home: current:=first;
    ed: current:=last;
end;(case)
end;(while)
end;(page)

```

```
    registro:=rtem;  
end;(borra_linea)
```

```
procedure inserta_linea;  
var  
k,p : integer;  
rtem: dato;
```

```
begin  
    rtem:=registro_ceros;  
    p :=ref+linea;  
  
    for k:=1 to p-1 do  
        begin  
            rtem.campo_01[k]:=registro.campo_01[k];  
            rtem.campo_02[k]:=registro.campo_02[k];  
        end;  
  
    for k:=p to numero_elementos-1 do  
        begin  
            rtem.campo_01[k+1]:=registro.campo_01[k];  
            rtem.campo_02[k+1]:=registro.campo_02[k];  
        end;  
  
    registro:=rtem;  
end;(inserta_linea)
```

```
procedure fija_registro_ceros;  
begin  
    registro_ceros.campo_01[ 1]:=' 0';  
    registro_ceros.campo_01[ 2]:=' 0';  
    registro_ceros.campo_01[ 3]:=' 0';  
    registro_ceros.campo_01[ 4]:=' 0';  
    registro_ceros.campo_01[ 5]:=' 0';  
    registro_ceros.campo_01[ 6]:=' 0';  
    registro_ceros.campo_01[ 7]:=' 0';  
    registro_ceros.campo_01[ 8]:=' 0';  
    registro_ceros.campo_01[ 9]:=' 0';  
    registro_ceros.campo_01[10]:=' 0';  
    registro_ceros.campo_01[11]:=' 0';  
    registro_ceros.campo_01[12]:=' 0';  
    registro_ceros.campo_01[13]:=' 0';  
    registro_ceros.campo_01[14]:=' 0';  
    registro_ceros.campo_01[15]:=' 0';
```

registro_ceros.campo_01[16]=' 0';
registro_ceros.campo_01[17]=' 0';
registro_ceros.campo_01[18]=' 0';
registro_ceros.campo_01[19]=' 0';
registro_ceros.campo_01[20]=' 0';
registro_ceros.campo_01[21]=' 0';
registro_ceros.campo_01[22]=' 0';
registro_ceros.campo_01[23]=' 0';
registro_ceros.campo_01[24]=' 0';
registro_ceros.campo_01[25]=' 0';
registro_ceros.campo_01[26]=' 0';
registro_ceros.campo_01[27]=' 0';
registro_ceros.campo_01[28]=' 0';
registro_ceros.campo_01[29]=' 0';
registro_ceros.campo_01[30]=' 0';
registro_ceros.campo_01[31]=' 0';
registro_ceros.campo_01[32]=' 0';
registro_ceros.campo_01[33]=' 0';
registro_ceros.campo_01[34]=' 0';
registro_ceros.campo_01[35]=' 0';
registro_ceros.campo_01[36]=' 0';
registro_ceros.campo_01[37]=' 0';
registro_ceros.campo_01[38]=' 0';
registro_ceros.campo_01[39]=' 0';
registro_ceros.campo_01[40]=' 0';

registro_ceros.campo_02[1]=' 0.00';
registro_ceros.campo_02[2]=' 0.00';
registro_ceros.campo_02[3]=' 0.00';
registro_ceros.campo_02[4]=' 0.00';
registro_ceros.campo_02[5]=' 0.00';
registro_ceros.campo_02[6]=' 0.00';
registro_ceros.campo_02[7]=' 0.00';
registro_ceros.campo_02[8]=' 0.00';
registro_ceros.campo_02[9]=' 0.00';
registro_ceros.campo_02[10]=' 0.00';
registro_ceros.campo_02[11]=' 0.00';
registro_ceros.campo_02[12]=' 0.00';
registro_ceros.campo_02[13]=' 0.00';
registro_ceros.campo_02[14]=' 0.00';
registro_ceros.campo_02[15]=' 0.00';
registro_ceros.campo_02[16]=' 0.00';
registro_ceros.campo_02[17]=' 0.00';
registro_ceros.campo_02[18]=' 0.00';
registro_ceros.campo_02[19]=' 0.00';

```

registro_ceros.campo_02[20]:=' 0.00';
registro_ceros.campo_02[21]:=' 0.00';
registro_ceros.campo_02[22]:=' 0.00';
registro_ceros.campo_02[23]:=' 0.00';
registro_ceros.campo_02[24]:=' 0.00';
registro_ceros.campo_02[25]:=' 0.00';
registro_ceros.campo_02[26]:=' 0.00';
registro_ceros.campo_02[27]:=' 0.00';
registro_ceros.campo_02[28]:=' 0.00';
registro_ceros.campo_02[29]:=' 0.00';
registro_ceros.campo_02[30]:=' 0.00';
registro_ceros.campo_02[31]:=' 0.00';
registro_ceros.campo_02[32]:=' 0.00';
registro_ceros.campo_02[33]:=' 0.00';
registro_ceros.campo_02[34]:=' 0.00';
registro_ceros.campo_02[35]:=' 0.00';
registro_ceros.campo_02[36]:=' 0.00';
registro_ceros.campo_02[37]:=' 0.00';
registro_ceros.campo_02[38]:=' 0.00';
registro_ceros.campo_02[39]:=' 0.00';
registro_ceros.campo_02[40]:=' 0.00';
end;(blanqueo_registro)

```

```

function menu:char;
var
  opcion : char;
begin
  repeat
    gotoxy (68,10);
    read (kbd,opcion);
    opcion := upcase (opcion);
  until opcion in ['A'..'E'];
  menu := opcion;
  write (opcion);
end;(menu)

```

```

procedure imprime_menu;
begin
  TextColor (green);
  GotoXY(52, 1);write('XXXXXXXXXXXXXXXXXXXXXXXXXXXX');
  GotoXY(52, 2);write(' :      BASE DE DATOS      ');
  GotoXY(52, 3);write('XXXXXXXXXXXXXXXXXXXXXXXXXXXX');
  GotoXY(52, 4);write(' :  A. CREA CURVA      ');
  GotoXY(52, 5);write(' :  B. EDITA CURVA    ');
  GotoXY(52, 6);write(' :  C. BORRA CURVA    ');

```

```

GotoXY(52, 7);write(' D. CATALOGO DE CURVAS ');
GotoXY(52, 8);write(' E. SALIR ');
GotoXY(52, 9);write('DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD');
GotoXY(52,10);write(' OPCION : _ ');
GotoXY(52,11);write('HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH');
end;(imprime_menu)

```

```

procedure mensaje(texto: msg);
begin
  TextColor(LightRed);
  GotoXY(24,25);
  write(texto, ' (ENTER) para continuar');
  repeat
    inkey(cod);
  until cod in[enter];
  GotoXY(24,25);write(' ');
end;(mensaje)

```

```

procedure confirma(texto: msg;var sino: boolean);
begin
  TextColor(LightRed);
  GotoXY(24,25);
  write(texto, ' s/n');
  repeat
    inkey(cod);
    if cod in[A..Z] then cod:=cod+32;(a minusculas)
  until cod in[s,n];
  GotoXY(24,25);write(' ');
  sino:=cod=s;
end;(confirma)

```

```

procedure titulos;
begin
  TextColor(green);
  GotoXY(1, 1);write('IMMMMMMMMMMMMMMMMM');
  GotoXY(1, 2);write(' I 3 t ');
  GotoXY(1, 3);write('DDDDDDDDDDDDDDDDDD');
  GotoXY(1, 4);write(' 3 : F1: GRABA ');
  GotoXY(1, 5);write(' 3 : F4: INSERTA LINEA ');
  GotoXY(1, 6);write(' 3 : F5: BORRA LINEA ');
  GotoXY(1, 7);write(' 3 : F6: ORDENA DATOS ');
  GotoXY(1, 8);write(' 3 : ESC: CANCELA ');
  GotoXY(1, 9);write(' 3 : BARRA: BLANQUEA CELDA ');
  GotoXY(1,10);write(' 3 : HOME: PRIMERA PAGINA ');
  GotoXY(1,11);write(' 3 : PgUp: PAGINA ANTERIOR ');

```



```

GotoXY(1,12);write('      3      : PgDn: PAGINA SIGUIENTE');
GotoXY(1,13);write('      3      : END: ULTIMA PAGINA  ');
GotoXY(1,14);write('      3      :');
GotoXY(1,15);write('      3      :');
GotoXY(1,16);write('      3      :');
GotoXY(1,17);write('      3      :');
GotoXY(1,18);write('      3      :');
GotoXY(1,19);write('      3      :');
GotoXY(1,20);write('      3      :');
GotoXY(1,21);write('      3      :');
GotoXY(1,22);write('      3      :');
GotoXY(1,23);write('      3      :');
GotoXY(1,24);write('#####XXXXXXXXXX');
end;(titulos)

```

```

(=====programa principal=====)
begin

(parametros)
columnas[1]:=3;
columnas[2]:=12;
fija_registro_ceros;

(carga de la base)
assign(file_var,'base.dat');
reset(file_var);
read(file_var,datos_disco);
max_registro:=datos_disco.ultima_curva;
datos:=datos_disco.datos;
close(file_var);

fin:=false;

01:
repeat
  clrscr;
  imprime_menu;
  case menu of
    (-----crea curva-----)
      'A' :
begin;
  (lee codigo y descripcion de la curva, verifica repeticion)
  if max_registro=numero_registros then begin mensaje('La base de datos esta llena');goto 01; end;

```

```

TextColor(LightBlue);
repeat
  gotoxy (24,23);
  write ('Codigo : ');
  readln (codigo_curva);
until codigo_curva<>copy(espacio,1,length(codigo_curva));
for i:=1 to max_registro do
begin
  if datos[i].codigo_curva=codigo_curva then begin mensaje('Ya existe este codigo'); goto 01; end;
end;
gotoxy (24,24);
write ('Descripcion: ');
readln (descripcion_curva);

(carga de datos)
registro:=registro_ceros;

(inicializa variables)
pagina:=1;
linea :=1;
campo :=1;

titulos;
despliegue_curva(ref);

repeat
  fila :=fila_referencia+linea;
  columna:=columnas[campo];

  case campo of
1: lee_entero (tecla,registro.campo_01[ref+linea]);
2: lee_decimal(tecla,registro.campo_02[ref+linea]);
  end;

  case tecla of
    f1: begin ordena_curva;repeticion; end;
    f4: begin inserta_linea;despliegue_curva(ref); end;
    f5: begin borra_linea;despliegue_curva(ref); end;
    f6: begin ordena_curva;repeticion;despliegue_curva(ref);tope; end;
    pg_up: begin pagina:=pagina-1;despliegue_curva(ref);tope; end;
    pg_dn: begin pagina:=pagina+1;despliegue_curva(ref);tope; end;
    home: begin pagina:=1;despliegue_curva(ref);tope; end;
    ed: begin pagina:=numero_paginas;despliegue_curva(ref);tope; end;
    enter: begin if campo=numero_campos then begin campo:=1;linea:=linea+1; end
              else campo:=campo+1;

```

```

        end;
    arriba: linea:=linea-1;
    abajo: linea:=linea+1;
    izquierda: begin if campo=1 then begin campo:=numero_campos;linea:=linea-1; end
                else campo:=campo-1;
    end;
    esc: begin
        confirm('cancela',conf);
        if not conf then tecla:=0;
    end;
end;
until tecla in {f1,esc};

(grabacion)
if tecla=f1 then
    begin
        registro.codigo_curva :=codigo_curva;
        registro.descripcion_curva:=descripcion_curva;
        max_registro:=max_registro+1;
        datos[max_registro]:=registro;
    end;

end;
(-----edita curva-----)
'B' :
begin;
    {lee codigo y descripcion de la curva, verifica existencia}
    if max_registro=0 then begin mensaje('La base de datos esta vacia');goto 01; end;
    TextColor(LightBlue);
    repeat
        gotoxy (24,23);
        write ('Codigo : ');
        readln (codigo_curva);
    until codigo_curva<>copy(espacio,1,length(codigo_curva));
    no_esta:=true;
    for i:=1 to max_registro do
        begin
            if datos[i].codigo_curva=codigo_curva then begin no_esta:=false;reg:=i; end;
        end;
    if no_esta then begin mensaje('No existe este codigo'); goto 01; end;
    gotoxy (24,24);
    write ('Descripcion: ',datos[reg].descripcion_curva);

    (carga de datos)
    registro:=datos[reg];

```

```

(inicializa variables)
pagina:=1;
linea :=1;
campo :=1;

titulos;
despliegue_curva(ref);

repeat
  fila :=fila_referencia+linea;
  columna:=columnas[campo];

  case campo of
1: lee_entero (tecla,registro.campo_01[ref+linea]);
2: lee_decimal(tecla,registro.campo_02[ref+linea]);
  end;

  case tecla of
    f1: begin ordena_curva;repeticion; end;
    f4: begin inserta_linea;despliegue_curva(ref); end;
    f5: begin borra_linea;despliegue_curva(ref); end;
    f6: begin ordena_curva;repeticion;despliegue_curva(ref);tope; end;
    pg_up: begin pagina:=pagina-1;despliegue_curva(ref);tope; end;
    pg_dn: begin pagina:=pagina+1;despliegue_curva(ref);tope; end;
    home: begin pagina:=1;despliegue_curva(ref);tope; end;
    ed: begin pagina:=numero_paginas;despliegue_curva(ref);tope; end;
    enter: begin if campo=numero_campos then begin campo:=1;linea:=linea+1; end
              else campo:=campo+1;
            end;
    arriba: linea:=linea-1;
    abajo: linea:=linea+1;
    izquierda: begin if campo=1 then begin campo:=numero_campos;linea:=linea-1; end
                else campo:=campo-1;
              end;
    esc: begin
            confirma('cancela',conf);
            if not conf then tecla:=0;
            end;
          end;
  end;
until tecla in{f1,esc};

{grabacion}
if tecla=f1 then
  begin

```

```

        registro.codigo_curva      :=codigo_curva;
        registro.descripcion_curva:=descripcion_curva;
        max_registro:=max_registro+1;
        datos[reg]:=registro;
    end;
end;

(-----borra curva-----)
    '0' :
begin
    (lee codigo y descripcion de la curva, verifica existencia)
    if max_registro=0 then begin mensaje('La base de datos esta vacia');goto 01; end;
    TextColor(LightBlue);
    repeat
        gotoxy (24,23);
        write ('Codigo : ');
        readln (codigo_curva);
    until codigo_curva<>copy(espacio,1,length(codigo_curva));
    no_esta:=true;
    for i:=1 to max_registro do
        begin
            if datos[i].codigo_curva=codigo_curva then begin no_esta:=false;reg:=i; end;
        end;
    if no_esta then begin mensaje('No existe este codigo'); goto 01; end;
    gotoxy (24,24);
    write ('Descripcion: ',datos[reg].descripcion_curva);

    (carga de datos)
    registro:=datos[reg];

    (inicializa variables)
    pagina:=1;
    linea :=1;
    campo :=1;

    titulos;
    despliegue_curva(ref);

    (eliminacion)
    confirma('Elimina la curva',conf);
    if conf then
        begin
            for k:=reg+1 to max_registro do
                begin
                    datos[k-1]:=datos[k];

```

```

procedure page(file_name:s7);

const
lpp=15;           (lineas por pagina)
bin_row = '
esc      =27;
pg_up    =195;
pg_dn    =203;
home     =193;
ed       =201;

type
cadena=string[70];
page_pointer='page;
page= record
  row01: cadena;
  row02: cadena;
  row03: cadena;
  row04: cadena;
  row05: cadena;
  row06: cadena;
  row07: cadena;
  row08: cadena;
  row09: cadena;
  row10: cadena;
  row11: cadena;
  row12: cadena;
  row13: cadena;
  row14: cadena;
  row15: cadena;
  pag  : string[3];
  prev : page_pointer;
  next : page_pointer;
end;

var
first,current,last: page_pointer;
j,k,n_pag,comm    : integer;
file_var          : text;
registro          : cadena;
row               : array[1..15] of cadena;

procedure nueva_pagina;
var

```

```

npag: string(3);

begin
  n_pag:=n_pag+1;
  str(n_pag,3,npag);

  new(current);
  with current^ do begin
    row01:=row[ 1];
    row02:=row[ 2];
    row03:=row[ 3];
    row04:=row[ 4];
    row05:=row[ 5];
    row06:=row[ 6];
    row07:=row[ 7];
    row08:=row[ 8];
    row09:=row[ 9];
    row10:=row[10];
    row11:=row[11];
    row12:=row[12];
    row13:=row[13];
    row14:=row[14];
    row15:=row[15];
    pag :=npag;
  end;(with)
  if first=nil then begin
    first=current;
    current^.prev:=nil;
    current^.next:=nil;
    last=current;
  end
  else begin
    last^.next:=current;
    current^.prev:=last;
    current^.next:=nil;
    last:=current;
  end;(if)
end;(nueva_pagina)

procedure show_page;
begin
  gotoxy(1,7);
  with current^ do begin
    writeln(row01);
    writeln(row02);

```



```

        writeln(row03);
        writeln(row04);
        writeln(row05);
        writeln(row06);
        writeln(row07);
        writeln(row08);
        writeln(row09);
        writeln(row10);
        writeln(row11);
        writeln(row12);
        writeln(row13);
        writeln(row14);
        writeln(row15);
        gotoxy(1,22);
        write('Pagina ',pag,'/',n_pag);
    end;(with)
end;(show_page)

procedure read_com;
var
ok: boolean;
begin
    repeat begin
        repeat
            inkey(com);
        until com in[esc,pg_up,pg_dn,home,ed];
        ok:=true;
        if (com=pg_up) and (current=first) then ok:=false;
        if (com=pg_dn) and (current=last ) then ok:=false;
        if (com=home ) and (current=first) then ok:=false;
        if (com=ed ) and (current=last ) then ok:=false;
    end;(repeat)
    until ok
end;(read_com)

(programa principal-----)
begin
    (load file-----)
    assign(file_var,file_name);
    reset(file_var);
    first:=nil;
    n_pag:=0;
    k :=0;

    if eof(file_var) then begin close(file_var);halt; end;

```