

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

**Automatización de las Empresas Eléctricas
por medio del Programa Cablecad AM / FM**

**Aplicación del Lenguaje de Programación de
Cablecad a un sistema de Distribución Eléctrica**

Tópico de Graduación

Previo a la Obtención del Título de

Ingeniero en Computación

Perteneciente a:

Stalin Raúl del Salto Montero

Guayaquil - Ecuador

1997

DEDICATORIA

A MI MADRE



Estas pocas letras mas que una dedicatoria es un homenaje para el ser más maravilloso del universo, para la persona que siempre estuvo presente y que hizo posible la culminación de mis estudios , a ti madre te debo todo lo que soy , quiero que sepas que siempre te llevo en mi corazón.

Te amo

Tu hijo

AGRADECIMIENTO

Un particular agradecimiento a la Escuela Superior Politécnica del Litoral, por haberme acogido en su Institución, y brindarme todos los conocimientos necesarios e importantes para poder desenvolverme en mi vida profesional. Para así de esa manera poder devolver en algo todo el tiempo que mis maestros dedicaron su tiempo y aportar con esto al engrandecimiento de mi país.

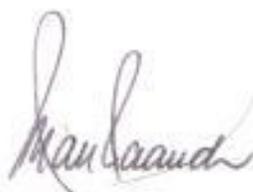
A los profesores, que con su paciencia y dedicación pusieron en nosotros la semilla del saber, para que esta pueda dar el mejor fruto y poner en muy alto el nombre de nuestra Institución. Y seguir manteniendo el buen nombre que se ha forjado.

Y al Ing. Juan Saavedra Mera quien supo guiarme, a paso firme para que mi trabajo reflejara esfuerzo y dedicación a través de sus sabios conocimientos.

MIEMBROS DEL TRIBUNAL



Ing. Altamirano
Sub-Decano de la Facultad
de Ingeniería en Electricidad
Y Computación



Ing. Juan Saveedra
Director del Tópico



Ing. Jorge Chiriboga
Vocal del Tribunal



Ing. Cristóbal Mera
Vocal del Tribunal

DECLARACION EXPRESA

“ La responsabilidad por los hechos, ideas expuestas en este informe de tópicos de graduación, corresponde exclusivamente al integrante mencionado a continuación y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral” .

Stalin del Salto

Stalin Raúl Del Salto Montero.

PROLOGO

En el capítulo 1 se realiza una breve introducción al conceptos familiares como son AM/FM GIS. Explicando características de cada uno de los conceptos y haciendo un breve análisis comparativo sobre ellos. Además de esto se realiza una explicación sobre el sistema CABLECAD.

En él capítulo 2 se explica los conceptos y elementos que una persona puede seguir para realizar una programación dentro de CABLECAD

En él capítulo 3 se explica los diferentes comandos utilizados en el desarrollo del sistema "CURSO".

En él capítulo 4 se explica el utilitario mas importante utilizado en la elaboración de los diferentes programas que componen el sistema "CURSO".

En él capítulo 5 se da una explicación de las diferentes funciones que componen el sistema "CURSO".

CONTENIDO E INDICE

Prólogo	1
Objetivo	1
CAPITULO 1	
Introducción a un mundo de información gráfica	2
1.1. Automatización de mapeo y facilidades de gerencia AM/FM	2
1.2. Información de sistemas gráficos GIS	3
1.3. AM/FM vs GIS	3
1.4. Introducción a OS2	4
1.5. Introducción a CABLECAD	5
1.6. Ambiente de trabajo	7
1.6.1. Requerimientos técnicos	7
CAPITULO 2	
Introducción a la programación en CABLECAD	8
2.1. Variables del Sistema	8
2.2. Funciones del Sistema	10
2.2.1. Comentarios	10
2.2.2. Expresión	11
2.2.3. Asignamientos	11
2.3. Lenguaje de comandos UDC	12
2.3.1. Como escribir una UDC	12
2.4. Compilación	14
CAPITULO 3	
Lenguaje de Programación	16
2.1. Resumen de los comandos	16
CAPITULO 4	
Utilitarios de CABLECAD	34
4.1. Utilitario GEN	34
CAPITULO 5	
Desarrollo del sistema	36
5.1. Menu "CURSO"	36
5.2. Definición de registros	43
Conclusiones	47
Recomendaciones	49

OBJETIVO

Es el de aplicar en forma didáctica el lenguaje de programación de CableCad para la creación de Base de Datos inteligentes que combinen el gráfico y el texto aplicado a un Sistema de Distribución Eléctrica.

La aplicación a desarrollarse se llama "CURSO" y el acceso se lo realiza a través del menú de pantalla

1.- INTRODUCCION A UN MUNDO DE INFORMACION GRAFICA (CABLECAD)

En un mundo como hoy la información gráfica es tan importante como la escrita, existen asociaciones y conceptos tales como AMIFM y GIS para ayudar a desarrollar esta información gráfica, para presentarla existen una diversidad de formas de hacerlo, una de ellas es un producto conocido con el nombre de CableCad el cual fue creado por la compañía Enghouse System de Canadá, este producto presenta una gran variedad de aplicaciones para diversas ramas de la electricidad, para este tópico CableCad esta orientado a la área de Distribución Eléctrica.

CableCad a parte de mostrar estas aplicaciones que permiten la fácil realización de dibujos, se presenta como un sistema abierto, en el cual se puede ingresar y ver el código fuente que puede ser alterado o crear uno nuevo si es necesario, definiendo de esta manera una UDC (un programa que al ser compilado se convierte en ejecutable), para ello el presente documento da una visión general sobre AMIFM\GIS y explica además los diferentes componentes que ayudaran a la creación de una UDC, para finalizar con un ejemplo aplicado al área de Distribución Eléctrica desarrollado sobre CableCad.

1.1. AUTOMATIZACION DE MAPEO Y FACILIDADES DE GERENCIA AM/FM

La misión de AM/FM es de proveer una excelencia en la educación y un intercambio de información en el uso y beneficios de la tecnología de información espacial para comunicación, infraestructura y la utilidad de aplicaciones mundiales.

AM/FM Internacional es una asociación educacional en el mundo que enfoca aplicando la cartografía automatizada y las facilidades de gerencia (AM/FM), y la información relacionada con las aplicaciones de tecnología para utilidades de agencias de Gobierno y otras organizaciones interesadas en registrar tierras y otras facilidades de gerencia.

Esto es un invaluable recurso para profesionales en muchas áreas quienes están con frecuencia incrementando el viaje para la información de tecnología de base de computadoras que ayudan a encontrar un análisis de datos que se necesite tanto local, regional, de estado, nacional y global.

AM/FM provee un foro para educadores expertos y novatos para hablar de procesos, y soluciones al estar accediendo y dirigiendo la cartografía y las facilidades de información de base de datos.

Los miembros llegan hacer una parte integral de una red que esta expandiendo y envolviendo todos los niveles y facetas de las industrias AM/FM.

1.2. INFORMACION DE SISTEMAS GRAFICOS GIS

En su forma más simple, el termino Información de Sistemas Geográficos GIS, es la sigla de la industria para una tecnología que ofrece una manera radicalmente diferente en que nosotros producimos y usamos los mapas necesitados para dirigir nuestras comunidades e industrias.

Usando los programas de computadoras, la tecnología enlaza los articulos mostrados con grabaciones (llamado atributos de datos) en una base de datos con las respuestas mostradas en un mapa. La combinación resultante, y la habilidad para manipular los datos en respuesta alguna petición de información se conoce como GIS. Existen un numero de agencias gubernamentales, y una larga lista de industrias privadas con una poderosa y dinámica herramienta que tiene abierta las puertas en la eficacia de gerencia y eficiencia de organización. Un GIS crea supermapas inteligentes por lo cual los planes sofisticados y análisis pueden ser ejecutados al tirar un pulsador.

1.3. AM/FM vs GIS

AM/FM es un subjuego de GIS que difiere de otras aplicaciones del mismo genero.

Un poco de características específicas y polígonos como características topológicas es común en algunas bases de datos de GIS, un modelo lineal de red constituye un AM/FM (ejemplo CableCad) que usa una base de datos inteligente para analizar y recorrer aplicaciones por características lineales tales como pipes o cables.

La cartografía o mapeo aplicada en ordenadores o computadoras, esta diseñada al manejo de redes de servicios, pero por su parte normalmente constan de un modulo de ayuda gráfica y es por tanto una aplicación netamente vectorial, lo cual se considera conveniente para este propósito.

AM/FM adicionalmente cuentan con manejadores y/o utilerías similares a los usados en GIS. Por el hecho de estar basados es una estructura vectorial, es menos eficiente que GIS en el manejo y relación de polígonos, pero es adecuadamente versátil, en la administración de elementos de red y su entorno.

Los productos AM/FM, utilizan referencias de posición, que son libres al usuario. Esto puede ser práctico para áreas pequeñas donde no es crítico conocer la localización geográfica. Además pueden ser trasladados, a sistemas X-Y relacionadas con la posición geográfica. Esto es muy importante para permitir trasladar información, utilizando referencias cartográficas. Normalmente no se usan referencias de tipo lat/lon en este tipo de productos, los cuales deben configurar si se desea.

1.4. INTRODUCCION A OS2

El sistema operativo que vamos a utilizar para el proyecto es el OS2 que es el ferviente competidor de WINDOWS. OS2 fue creado por la IBM, mientras que WINDOWS fue creado por MICROSOFT.

En la instalación de OS2 a la PC hace una partición lógica en el BIOS, la cual divide al disco duro en dos partes denominadas C y D, en este caso en la parte C se asigna el DOS y en la parte D se la asigna al sistema operativo OS2.

Este sistema operativo soporta también DOS y OS programas en WINDOWS creando su ventana de trabajo WIN-OS2, este sistema operativo también tiene su lenguaje nativo que es el OS2 que es muy parecido al DOS.

1.5. INTRODUCCION A CABLECAD

CableCad fue un sistema diseñado para satisfacer las necesidades de información gráfica, su utilidad radica en sistemas de información a todo nivel.

Es una herramienta de tipo AM/FM GIS que puede ser utilizado para redes como por ejemplo las redes Eléctricas, redes Telefónicas, redes de Aguas Servidas, etc. , o en cualquier aplicación donde la información gráfica sea necesaria.

CableCad le permite trabajar con Tablas de Validación, Definición de Registros, UDC, Definición de Símbolos, Definición de Menús.

En las Tablas de Validación se encuentra parámetros contra los cuales se validan los ingresos de datos, los datos ingresados son comparados y de este modo verificar su integridad, estos parámetros se encuentran en los archivos ENGEN.DAT y ENGEN.IDX

Posee además un esquema de Definición de Símbolos, el cual permite crear símbolos y darle ciertas características importantes (color, tamaño, etc..) estos pueden ser creados mediante el SYMGEN.

Cuenta además con una estructura general de archivos los cuales pueden ser clasificados de la siguiente manera:

*. GRF	Archivos Gráficos
*. NGF	Archivos No Gráficos
*. ATB	Archivos de Atributos
*. IDX	Archivos de Indice
*. QUD	Archivos de Quad Tree

Posee una Base de Datos Gráfica la cual posee información referente a líneas, puntos, texto impreso, símbolos y demás elementos. Son elementos visuales que al igual que en un plano, permiten al usuario contar con puntos de referencia visual o información gráfica necesarias para la interpretación de la información.

Estos elementos tienen propiedades, estas pueden ser modificadas, dando un mayor control a cualquier tipo de aplicación que se quiera realizar de una manera muy simple.

Esta parte de información se ve complementada por una Base de Datos no Gráfica, la cual posee información sobre las características especiales de los elementos, estas son grabadas en registros, que pueden ser modificados o creados en el caso de no existir

Los elementos gráficos son administrados y manejados a través del menú de BasePlan en el menú principal de CableCab.

Estos dos ambientes gráficos y no gráficos son soportados por medio de un macrolenguaje de programación el cual es dúctil permitiendo una programación estructurada al usuario por medio de sentencias y funciones definidas por CableCad, aquellas funciones que no fueron definidas pueden ser creadas por medio de este lenguaje, el realizar este proceso se conoce con el nombre de UDC, las cuales pueden ser compiladas convirtiéndose en ejecutables con extensión UCC.

Al juntar todos estos elementos mencionados otorgan a CableCad una característica muy importante, que es la de convertirse en un gráfico inteligente de este modo los dibujos presentados no solo muestran una característica gráfica, si no que además de esta presentan la característica de texto, esto significa que los gráficos poseen información de cualquier tipo que ayudan a la comprensión de un determinado mapa haciendo este aun más relevante.

1.6. AMBIENTE DE TRABAJO

CableCad necesita de un ambiente de trabajo propicio para poder desenvolverse de la mejor manera, este ambiente se lo conoce como requerimientos técnicos.

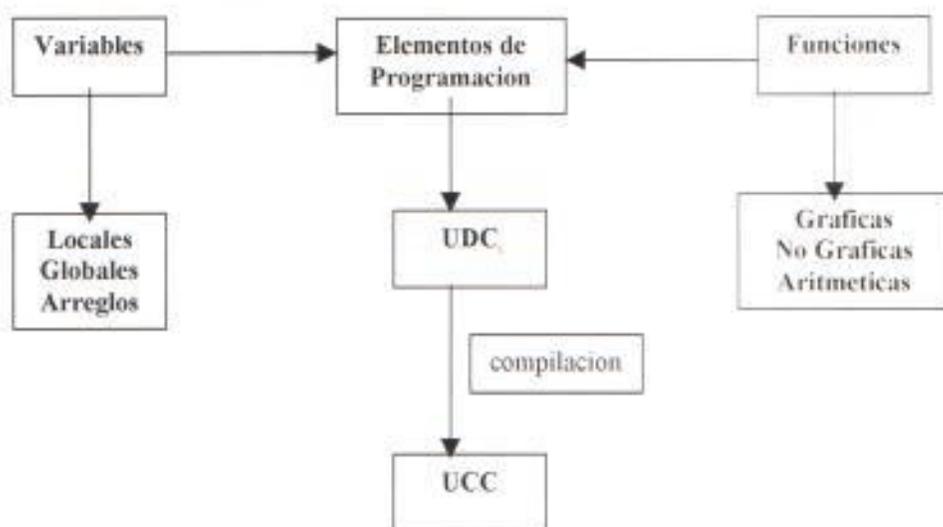
1.6.1. Requerimientos Técnicos

El equipo utilizado en el proyecto es el siguiente.

- CPU 16 MB. RAM.
- Procesador Pentium.
- Monitor de 14".
- Disco Duro de 1 GB.
- Tarjeta de Video SVGA.
- Sistema operativo: OS2.

2.- INTRODUCCION A LA PROGRAMACION EN CABLECAD

A continuación se muestran un gráfico explicativo de los distintos elementos que intervienen en la creación de un programa.



2.1.5 VARIABLES DEL SISTEMA

Posee información que puede ser asignadas por el usuario o seteadas por CableCad, las variables propias del sistema están precedidas por el signo \$.

Ejemplo:

\$abort: es un indicador de aborto de las UDC

Las variables pueden ser :

LOCALES:

Estas variables mantienen información mientras la UDC se encuentra activa, con la particularidad que su alcance se ve limitado a la UDC que se encuentra en ejecución en ese momento.

Estas variables pueden ser de tipo:

LINE
LSEG
NUMERIC
TEXT

Ejemplo:

Se antepone el signo @

@cantidad

@costo

GLOBALES:

Mantienen información a un después que la UDC ha finalizado su ejecución, de esta manera la información puede pasar de una UDC a otra.

Estas variables pueden ser de tipo:

LINE
LSEG
NUMERIC
TEXT

Ejemplo:

Se antepone el signo @

@cantidad

@costo

ARRAY:

Este tipo de variable puede ser local o global, pero con la particularidad de que su tipo es numérico

Ejemplo:

LOCAL (@arreglo[5], 'NUMERIC')

GLOBAL(@arreglo[10], 'NUMERIC')

Nota: Lo encerrado entre paréntesis es la dimensión del arreglo.

2.2. FUNCIONES DEL SISTEMA

Existen tres tipos de funciones gráficas, no gráficas, aritméticas. Estas generalmente pueden retornar un valor cuando son ejecutadas, las funciones pueden ser usadas como argumentos en ciertas instrucciones y pueden operar como asignamientos o expresiones.

Las funciones gráficas son utilizadas para manipular toda la información gráfica existente en CableCad, las no gráficas como su nombre lo indica se encargan de la manipulación de la información no gráfica, mientras que las aritméticas se encargan de la realización de cálculos.

Las funciones de CableCad comienzan con el símbolo de porcentaje % y pueden ser tratadas como operandos. A una función se le puede ingresar una expresión y retornar un valor.

Cuando Ud. usa una función como un operando Ud. llama a las funciones de CableCad para evaluar y utiliza el resultado como un operando en la expresión.

Ejemplo:

% abs, es una función propia de CableCad

HERRAMIENTAS:

En herramientas se encuentran ciertas utilidades que ayudan a una mejor comprensión de las funciones o facilidades que le permitan obtener un mejor desempeño en la creación de las mismas.

2.2.1. Comentarios

Son líneas de texto que me permiten dar una mejor descripción del programa.

Ejemplo:

@y = valor /* la variable @y almacena el costo del producto */

2.2.2. Expresión

Una expresión es un valor que consiste de dos o más constantes o variables separadas por un operador, este es un símbolo que denota algún tipo de operación como suma, resta, división, multiplicación o exponenciación.

Las constantes y variables asumen varias formas :

- Sistema de variables cuando acarrea un valor numérico
- Archivos de Base de Datos no repetidos.
- Constantes numéricas.
- Variables locales, globales o arreglos

Se debe respetar el uso de los paréntesis, y los operadores son los siguientes:

^	exponenciación
*	multiplicación
/	división
+	suma
-	resta

Las expresiones son evaluadas en un orden definido.

()	paréntesis
^	exponente
* and /	multiplicación y división
+ and -	suma y resta
"_>"	desde la izquierda a la derecha

2.2.3. Asignamientos

Un asignamiento es un mecanismo empleado para que una variable tome un valor determinado.

En CableCad se utiliza dos operadores separados por un símbolo igual (=). El operador de la izquierda (generalmente usado como variable) recibe el valor que se encuentra en la derecha, este puede ser un elemento, expresión o un

valor numérico. En el caso que se requiera realizar una asignación tipo texto estas serán asignadas utilizando instrucciones como TEXTCOPY Y TEXTFORMAT.

En CableCad el operador que puede estar en el lado izquierdo son los siguientes:

- Variable tipo numérico definida por el usuario.
- Archivos de Base de Datos no repetidos.
- Ciertas variables del sistema.

Ejemplo:

```
(valor -1)/@distancia
costo=@valor
@longitud=(10 + %cos($angle))*3
IF ( costo<@valor/10 )
  return()
ENDIF
```

2.3. LENGUAJE DE COMANDOS UDC

Es un grupo de instrucciones que le permite realizar tareas, las instrucciones pueden ser conjuntos de sentencias, variables, funciones o comandos .

La UDC puede estar escrita en un editor cualquiera, cuando esta se termina de escribir, esta puede ser compilada y convertirse en ejecutable (UCC).

2.3.1. Cómo escribir una UDC?

- 1.- En la primera sección del programa se declara las variables locales y/o globales.
- 2.- En la segunda sección defina los tipos de registros que van hacer utilizados.

- 3.- Mostrar el submenú (opcional).
- 4.- Escribir el programa, en esta sección se encuentra las instrucciones para la manipulación gráfica y no gráfica.
- 5.- Guarda la información de los archivos no gráficos.
- 6.- Se remueven los menús.

Ejemplo:

```
/* Archivo          PRUEBA.UDC
   Fecha de creación 99/99/99
   Programador      SDS
   Descripción      Ejemplo de como escribir una UDC */
```

/ Sección 1 Declaración de variables */*

```
GLOBAL (@menu,'TEXT')
```

/ Sección 2 Definición de registros */*

```
SETDBREC(POLE)
```

/ Sección 3 Mostrar el lugar donde se encuentra el menú */ (opcional)*

```
SETMENUNBR('\DIRECTORY\@menu',19)
```

/ Sección 4 Procesamiento */*

En esta sección se escribe el código que compone el programa

/ Sección 5 Información de no gráfica */*

```
WRITEDBREC('A',")
```

/ Sección 6 Remover menús */*

```
REMOVEMENUS
```

2.4. COMPILACION

Un factor importante de una UDC cuando esta termina es el poder realizar una ejecución de la misma, por medio del utilitario CLCOMP.EXE le permite compilar , o convertir , una UDC en un formato que pueda ser entendido por CABLECAD, una vez realizado este proceso el programa queda con extensión UCC.

SINTAXIS:

CLCOMP nombre[.UDC] [BANDERA]

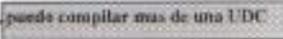
La **BANDERA** se encarga de realizar cierta particularidad, antes o después de realizar el proceso de compilación.

TIPOS DE BANDERAS:

/D	Prepara una ruta para el código
/H	Llamá a la ayuda de Window
/L userid [password]	Login especifico para un usuario
/NC	No chequea los argumentos de los comandos
/R	Genera una referencia rápida del comando

Caracteres como (?,*) pueden ser usados cuando usted especifica el nombre de la UDC.

Si desea compilar varias UDCs, puede proceder de la siguiente manera.

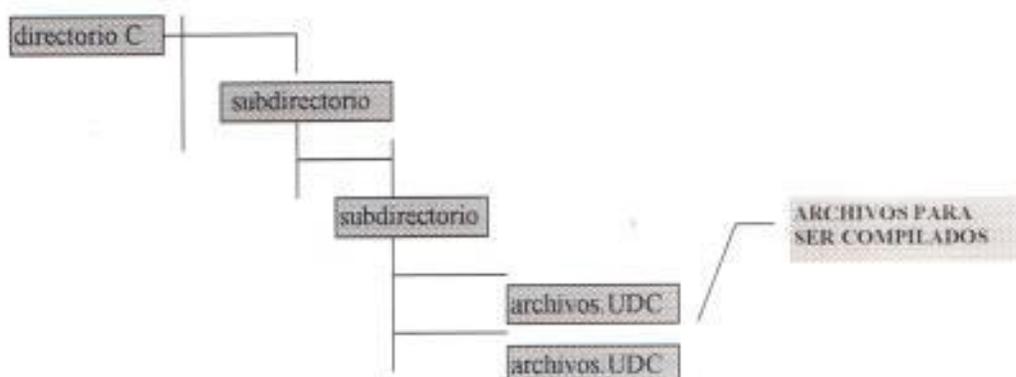
CLCOMP * 

Si desea compilar una UDC en un directorio especifico(supongamos que el directorio de llama tópico), puede proceder de la siguiente manera.

CLCOMP \directorio\ subdirectorio\ topico\nombre.udc

Si desea compilar todas la UDCs de un directorio en específico (supongamos que el directorio de llama tónico).

CLCOMP \directorio\ subdirectorio\ subdirectorio\ topico*



3. LENGUAJE DE PROGRAMACION

A continuación se explicara los distintos comandos utilizados para el desarrollo de la base de datos gráfica y no gráfica aplicado a un sistema de Distribución Eléctrica.

3.1. RESUMEN DE LOS COMANDOS

Se presenta un resumen de comandos :

FLDGET	Ingresar datos
GDRAW	Dibujar item
PLCSYMBOL	Posicionar un símbolo
READBREC	Lee información
REMOVEMENUS	Remueve un menú
RETURN	Realiza una interrupción en la ejecución de una UDC
SETDBREC	Establece un tipo de grabación
SETGCOLOR	Setea el color
SETFCODE	Setea el código
SETGSTYLE	Setea un estilo lineal existente
SETGLEVEL	Setea el nivel actual del gráfico
SETMENUNBR	Muestra el menú para la actual UDC
SETSVMNBR	Permite una colocación subsecuente
TEXTCOPY	Copia una cadena de caracteres a otra locación de memoria
WRITEDBREC	Crea inteligencia

FLDGET:

DESCRIPCION:

Esta instrucción permite ingresar datos desde teclado, y almacenarlo en el campo respectivo.

Nota: Los datos ingresados deben de ser del mismo tipo de datos creados en el SCRIPT

SINTAXIS: FLDGET(' prompt ', ' field ')

prompt : Especifica el texto

field : Especifica el campo

CODIGO DE RETORNO: FLDGET no regresa un valor \$status

Ejemplo: Seleccionar un poste e ingresar su peso.

FLDGET (' Seleccione el poste ', peso)

GDRAW:

DESCRIPCION:

Esta instrucción permite que el usuario dibuje ítem lineales, tales como líneas de alcantarilla, servicios de cable(líneas de cable), conductos, etc. En la terminación de cada operación GDRAW usa una instrucción SAVELINNE o WRITEDBREC al surtir la línea de dibujo en la base de datos gráfica. WRITEDBREC me permite establecer una conectividad con los ítems de la base de datos no gráfica, mientras SAVELINE me permite grabar esta información .

Nota: No use GDRAW con opción WRITEDBREC " C " use la opción WRITEDBREC " L "

SINTAXIS: Esta instrucción no tiene argumentos

CODIGO DE RETORNO: GDRAW no regresa un valor \$status, pero Ud. puede determinar el éxito de una operación GDRAW usando la variable \$nbrpnts. Si \$nbrpnts < 2, la línea no fue dibujada . Si \$nbrpnts > 1, la próxima instrucción WRITEDBREC surte la línea GDRAW .

Ejemplo: Permite al usuario dibujar una línea y grabarla.

```
GDRAW ( )  
SAVELINE ( )
```

Nota:

Cuando se guarda una línea GDRAW con la instrucción WRITEDBREC " L " , conectivamente queda establecido lo siguiente:

Si se realiza un snap (seleccionar) al comienzo del punto de la polilínea a un ítem de base de datos no gráfica, el ítem llega a ser padre de la polilínea.

Si se realiza un snap (seleccionar) al final del punto de la polilínea a un ítem de base de datos no gráfica, el ítem llega a ser hijo de la polilínea.

Si se realiza un snap (seleccionar) en un punto intermedio de la polilínea no se establece conectividad con ningún ítem de la base de datos.

PLCSYMBOL:

DESCRIPCIÓN:

Esta instrucción permite colocar un símbolo en el mapa de CableCad. Con esta instrucción se puede especificar la localización y la manera de colocación, pero no se puede especificar el símbolo a ser colocado. Se utiliza una instrucción SETSYMBOL delante de PLCSYMBOL para especificar el símbolo.

SINTAXIS: PLCSYMBOL('B', 'MENSAJE', 'O')

' B ' : Permite al usuario utilizar el botón de seleccionar

' O ' : Determina el ángulo del símbolo

CODIGO DE RETORNO: PLCSYMBOL no regresa un valor \$status

Ejemplo: Identifica el poste de teléfono y espera que el usuario presione snap button.

SETSYMBOL(27)

PLCSYMBOL('B', ' Presione el mouse ', ' O ')

READDBREC :

DESCRIPCIÓN:

Esta instrucción permite buscar y leer información no gráfica, esto sucede al leer el archivo de proyecto de grabaciones no gráficas (NGF). Esta instrucción depende de instrucciones precedentes al identificar el ítem o ítems a buscar para una siguiente instrucción al especificar el proceso ejecutado en ellos después de ser leído.

READDBREC puede encontrar información no gráfica en la siguiente manera.

- Para buscar un particular registro no gráfico numérico.
- Para buscar el actual archivo NGF. consecutivamente.
- Para buscar un índice de un registro no gráfico numérico.

Para buscar un índice de un particular registro.

Nota: Si se decide a realizar una búsqueda por índice este debe ser construido primero para esto de utiliza la instrucción BUILDINDEX.

Intercambiando entre modificar y crea modos (esto es mezclado las instrucciones READDBREC y SETDBREC) es permisible, pero siempre ejecuta

una operación WRITEDBREC antes de intercambiar modos o guardar modificaciones.

Cuando su instrucción READDBREC esta leyendo un índice (en este momento sé esta usando el tercero y cuarto argumento) el puntero del archivo indice es cambiado cada vez que la UDC ejecuta una instrucción WRITEDBREC. Al continuar leyendo la próxima secuencia del registro no gráfico desde el punto donde WRITEDBREC fue ejecutado , se debe resetear el archivo indice del puntero con una instrucción READDBREC (0, ' E ', ..., ...). Hacer esto es tan frecuente como necesario establecer un punto de inicio para una secuencia de la operación leer índice.

SINTAXIS: READDBREC(<record#>,[' E ' / ' F ' / ' N '] [,index] [,field]...)

record# : Especifica el actual registro no gráfico numérico, usualmente el registro a ser leído. Un valor cero para este argumento indica un reset al punto de inicio.

' E ' : READDBREC al ejecutar una operación " get equal " esto es leer un registro partiendo de la actual grabación no gráfica .

' F ' : READDBREC lee el primer registro no gráfico dependiendo de los valores seteados en el primer argumento .Si el primer argumento es seteado a cero, READDBREC retorna el primer registro en el archivo. Si el primer registro es un archivo no gráfico numérico valido, READDBREC lee dicho registro. Si este valor es no valido, READDBREC setea \$status a 1.

' N ' : READDBREC para leer el próximo registro secuencial no gráfico basado en el actual juego de valores seteados en el primer argumento.

index : Especifica el nombre (esto es el nombre del registro) de un índice, se construye previo a READDBREC con una instrucción BUILDINDEX.

filed : Especifica el nombre de cada campo esto es para ser usado como una llave de búsqueda en el índice especificado en el argumento.

Nota: Si se decide permitir a READBREC buscar por índice, se debe construir primero el índice con la instrucción BUILDINDEX.

CODIGO DE RETORNO:

Si READBREC es usado sin la opción índice, este setea \$status de la siguiente manera.

- 0 : Registro leído satisfactoriamente
- 1 : Irrecuperable I/O error o error en argumentos.
- 2 : Grabación específica que ha sido borrada.
- 99 : Debajo de la operación "N", no más grabaciones.

Si READBREC es usado con la opción de índice este setea \$status de la siguiente manera.

- 0 : Registro leído satisfactoriamente
- 1 : Irrecuperable I/O error o error en argumentos.
- 2 : Error en uno o más argumentos.
- 3 : No usado.
- 4 : Error en la llave, longitud o formato
- 5 : Llave de descripción no encontrada.
- 6 : Llave no encontrada durante la operación "E"
- 99 : Debajo de la operación "N", no más grabaciones.

Ejemplo:

- 1.- Seleccione un ítem y lee el ítem no gráfico de un registro.

```
SELECT('Select un ítem ' )  
IF ($button<> 3)  
    READBREC($ngfrechnbr)  
ENDIF
```

- 2.- Provee maneras alternativas para lectura de registros no gráficos en el índice del CUSTUMER, que usa el campo CONTRACT-NUMBER con una con una llave registrada . Entonces, lea la próxima secuencia de grabación.

```
READDBREC (@ngf, ' F ',CUSTUMER, CONTRACT-NUMBER )
WRITEDBREC ( ' M ', '' )
READDBREC ( ' O ', ' E ',CUSTUMER, CONTRACT-NUMBER )
READDBREC (@ngf, ' N ',CUSTUMER, CONTRACT-NUMBER )
```

3.- Asumiendo que se tiene que crear un índice POLE

```
READDBREC ($ngfrecnbr, ' E ',POLE)
IF $status =0
    READDBREC ($ngfrecnbr, ' N ',POLE)
ENDIF
```

REMOVEMENUS:

DESCRIPCIÓN:

Esta instrucción permite remover un menú de la pantalla de CableCad. Procede REMOVEMENUS con una instrucción SETMENUNBR, que identifica que menú ha sido removido.

SINTAXIS: Esta instrucción no tiene argumentos.

CODIGO DE RETORNO: REMOVEMENUS no regresa un valor \$status.

Ejemplo: Muestra una pieza del menú para proveer al usuario algunas opciones de línea-modificación.

Después que la línea ha sido modificada, se procede a remover el menú.

```
TEXCAT (@temp, ' \MENUS.OB' )
SETMENUNBR (@temp,29 )
MODIFYLINE ( )
REMOVEMENUS ( )
```

RETURN:

DESCRIPCIÓN:

Esta instrucción permite realizar una interrupción en la ejecución de una UDC donde la instrucción RETURN ocurre, el control retorna al programa principal de CableCad.

Un uso común de esta instrucción es la interrupción de la ejecución en una rama condicional a una condición error .

Si se omite RETURN como instrucción ultima en una UDC, la ejecución para de cualquier manera.

SINTAXIS: Esta instrucción no tiene argumento.

CODIGO DE RETORNO: RETURN no regresa un valor \$status.

Ejemplo: Probar un error de condición .

```
IF ( $status <> 0)
    MESSAGE ('Error condición no encontrada')
    RETURN ( )
ENDIF
```

SETDBREC:

DESCRIPCIÓN:

Esta instrucción permite establecer el tipo de grabación no gráfica para la actual y las siguientes instrucciones permite además poner la UDC en modo crear.

CableCad limita a uno él numero de registros activos, esto lo realiza en algún momento. Por lo tanto solo un buffer no gráfico esta disponible, esto significa que a cada hora(tiempo) se utiliza esta instrucción, desplazando el

previo SETDBREC. Desempeña una operación WRITEDBREC para salvar el buffer no gráfico.

Se debe tener conciencia que SETDBREC no setea la variable del sistema \$ngfrecnbr. Esta es seteada solamente cuando SETDBREC esta permitida por un WRITEDBREC, que finaliza un modo crear. Al final del modo crear, WRITEDBREC adjunta una grabación no gráfica al archivo (NGF) asumiendo que no interviene la operación READDBREC.

En general se especifica el tipo de grabación en el argumento de SETDBREC, este puede ser un nombre, tales como CABLE o POSTE. Se debe tener conciencia sin embargo, que si se utiliza una variable definida por el usuario como argumento de SETDBREC, se debe usar el tipo de registro numérico, si se setea el tipo de registro nombre en una variable de usuario causa un error.

SINTAXIS: SETDBREC (rectype)

rectype : Especifica el nombre del registro no gráfico

CODIGO DE RETORNO: SETDBREC no regresa un valor \$status.

Ejemplo: Setear el actual registro no gráfico a cable.
SETDBREC(CABLE)

SETGCOLOR:

DESCRIPCIÓN:

Esta instrucción permite setear el color del gráfico existente, las cuales pueden ser polilneas, símbolos, indicadores y texto gráfico (pero no el texto de la base de datos). SETGCOLOR sobrescribe la definición en el SCRIPTS para la duración de la UDC.

SINTAXIS: SETGCOLOR (color)

color : Especifica el color que puede ser usado de aquí en adelante por la UDC o hasta la próxima instrucción SETGCOLOR

CODIGO DE RETORNO: SETGCOLOR no regresa un valor \$status.

Ejemplo: Setee el gráfico a color verde para la duración de la UDC.
SETGCOLOR (GREEN)

SETGFCODE:

DESCRIPCIÓN:

Esta instrucción permite setear el código para objetos de gráficos existente al hacer una configuración de este. Estas gráficas pueden ser polilíneas, símbolos, indicadores y texto gráfico (pero no el texto de la base de datos). SETGFCODE sobrescribe el código de definición en SCRIPTS para la duración de la UDC.

SINTAXIS: SETGFCODE (level)

level : Especifica un valor desde 0 hasta 32767

CODIGO DE RETORNO: SETGFCOLOR no regresa un valor \$status.

Ejemplo: Setee el gráfico característico del código 1063 para la duración de la UDC:

SETFCODE (1063)

SETGSTYLE:

DESCRIPCIÓN:

Esta instrucción permite arreglar el gráfico de estilo lineal existente. El gráfico de estilo lineal es la apariencia física de las líneas de la gráfica de objetos, esta definido dentro de las utilidades SCRIPTS. Estas gráficas pueden ser de polilíneas, símbolos, indicadores y texto gráfico (pero no el texto de la base de datos); note que la presentación del texto de base de datos no es afectada por esta instrucción.

SETGSTYLE sobrescribe la definición en el SCRIPTS para la duración de la UDC

CableCad reorganiza un poco el estilo de nombre o el estilo de número como el argumento de esta instrucción.

SINTAXIS: SETGSTYLE (style)

style : Especifica el estilo de línea que debe usarse mientras dure la UDC o hasta la próxima instrucción SETGSTYLE

CODIGO DE RETORNO: SETGSTYLE no regresa un valor \$status.

Ejemplo: Setee el estilo de línea gráfica para la duración del UDC:

SETGSTYLE (MDASH)

SETGLEVEL:

DESCRIPCIÓN:

Esta instrucción permite setear el nivel actual de gráficos de objetos para un valor designado. Estos gráficos pueden ser de polilíneas, símbolos, indicadores

y texto gráfico (pero no el texto de la base de datos). SETGLEVEL sobrescribe la definición de los niveles en SCRIPTS para la duración del UDC o hasta la próxima instrucción SETGLEVEL.

SINTAXIS: SETGLEVEL (level)

level : Especifica un valor desde 0 hasta 32767 , como un nivel temporal para gráfica de objetos.

CODIGO DE RETORNO: SETGLEVEL no regresa un valor \$status.

Ejemplo: Setee el nivel gráfico de 23 para la duración del UDC :
SETGLEVEL(23)

SETMENUNBR:

DESCRIPCIÓN:

Esta instrucción permite arreglar y mostrar un menú para la actual UDC. Se puede proveer una ruta de especificación como parte de este primer argumento; además SETMENUNBR observa para el archivo de menú en el directorio seteado en la actual sesión para la mas reciente instrucción SETDIR.

Si esto no sucede SETMENUNBR observa en el siguiente archivo que se encuentra en directorio \CableCad\DSPIENGHOUSE.INI. Si SETMENUNBR no puede encontrar el archivo de titulo especifico, este presenta un mensaje erróneo y termina la UDC.

SINTAXIS: SETMENUNBR(file,menu#)

file : Especifica el nombre(y la ruta opcional) del archivo tile contiene el menú que puede ser mostrado.

menu# : Especifica el desplazamiento entre el archivo tile (primer menú es numero 0).

Nota: Archivo tile puede contener hasta 999 menús.

CODIGO DE RETORNO: SETMENUNBR no regresa un valor \$status.

Ejemplo: Muestre el menú 14avo en el archivo tile MENUS.OB :
SETMENUNBR (MENUS.OB, 3)

SETSYMNBR:

DESCRIPCIÓN:

Esta instrucción permite identificar un símbolo para una colocación subsecuente.

SINTAXIS: SETSYMNBR(id).

CODIGO DE RETORNO: SETSYMNBR no regresa un valor \$status.

Ejemplo: Poner el nombre del símbolo para una utilidad del poste.
SETSYMNBR(POLE)

TEXTCOPY:

DESCRIPCIÓN:

Esta instrucción permite copiar una cadena de caracteres a otra locación de memoria.

El valor del segundo argumento se sobrescribe al primer argumento.

El texto de destinación que provee una meta puede ser el siguiente.

- Una variable definida por el usuario
- Un campo de la base de datos no repetido
- Una constante definida como texto

SINTAXIS: TEXTCOPY (dest,source)

dest: Es el destino

SOURCE: Especifica el texto que se desea copiar.

CODIGO DE RETORNO: RETURN no regresa un valor \$status.

Ejemplo: Copiar un valor desde el prompt a una variable

TEXTCOPY (@temp, 'Entre un valor.')

WRITEDBREC:

DESCRIPCIÓN:

Esta instrucción permite crear inteligencia. Los gráficos inteligentes creados en una UDC son temporalmente almacenados en un buffer no gráfico. Use la instrucción WRITEDBREC para escribir desde el buffer el contenido a un archivo.

Asegurar que el procesamiento necesario de la grabación actual no gráfica se complete antes de la emisión de una instrucción WRITEDBREC, cada secuencia WRITEDBREC sobrescribe el buffer.

La UDC debe realizar un seteo para crear el modo antes de la instrucción WRITEDBREC que pueda ser efectivo. Crear el modo para empezar con una instrucción SETDBREC y usualmente finaliza con la instrucción WRITEDBREC.

Notas:

- 1.- Si se desea partir la polilinea de cada lado del símbolo insertado, use la opción WRITEDBREC ' C '
- 2.- Poniendo un símbolo al empezar o terminar un punto de una línea existente con la opción WRITEDBREC ' B ' dejar el espacio del símbolo principal cuando dividan la línea con su punto final. Si no se desean diferencias cuando pongan un símbolo al empezar o terminar polilinea, use la opción WRITEDBREC ' C ' al conectar o guardar el símbolo.
- 3.- Use la opción WRITEDBREC ' C ' solamente con la instrucción PLCSYMBOL .

Consideraciones GDRAW:

La instrucción GDRAW es una función de polilinea. Porque las polilíneas son usadas para dibujar la distribución de la red CableCad, las materias de la red llegan a ser importantes cuando se construyen polilíneas. Se ha mencionado arriba, que use una instrucción WRITEDBREC ' L ' cuando escriba las grabaciones de polilíneas a la base de datos y cuando se desea conectar cualquiera de los dos finales de una polilinea a símbolos no gráficos.

Enghouse no recomienda el uso de la opción WRITEDBREC ' C ' cuando se utiliza la instrucción GDRAW.

Opciones:

' A ' : Seleccionando el punto al comienzo o al final de la polilinea creada a un ítem no gráfico no lo establece conectivamente.

' B ' : Esta opción no opera con GDRAW

- ' L ' : Esta opción permite conectividad también del final de una polilínea. Use esta opción cuando se desea sujetar (atribuir) uno o ambos finales de una polilínea o símbolos.
- ' P ' : La grabación ha escrito un archivo no gráfico llegará a ser un padre de la grabación seleccionada para la más reciente instrucción SELECT. Conectivamente entre la polilínea creada y los ítems no gráficos no han sido establecidos.

SINTAXIS: WRITEDBREC (type,record) Formato General

Use la instrucción ' N ' y ' S ' para sujetar uno de los finales de la actual polilínea a otra polilínea

Formato 1: WRITEDBREC (N/S,record)

TYPE:

- ' N ' : Se crea un registro a un punto específico pero no se divide en dos la polilínea.
- ' S ' : Se crea un registro a un punto específico pero se divide en dos la polilínea.

record : Especifica el registro.

Formato 2: WRITEDBREC (' A ' , ' B ' , ' C ' , ' L ' , ' M ' , ' P ')

TYPE:

- ' A ' : Escribe un Buffer de un archivo no gráfico sin conectar el objeto debajo de la construcción de un ítem no gráfico
- ' B ' : No crea un registro en un punto específico, pero rompe la polilínea, en dos polilíneas y inserta un nuevo objeto entre ellos.
- ' C ' : No rompe la línea existente, pero insertar el nuevo objeto como un producto del ítem existente no gráfico en el punto específico.

- ' L ' : Si el nuevo objeto en el punto de comienzo esta seleccionado para la existencia de la grabación no gráfica, el nuevo objeto llega a ser un hijo del registro existente.

Si el nuevo objeto esta en el punto final esta seleccionado para la existencia de la grabación no gráfica, el nuevo objeto llega a ser un padre de la grabación existente.

- ' M ' : Para modificar el actual registro no gráfico. Asegurarse que se modifique el modo con la instrucción READDBREC antes de usar la instrucción WRITEDBREC.

WRITEDBREC escribe el buffer modificado a un archivo no gráfico.

- ' P ' : Añade un nuevo objeto como padre de un registro existente no gráfico en el punto específico.

Ejemplos:

1. Opción ' A ' :

```
SETDBREC (ANCHOR)
```

```
PLCSYMBOL ('S ','Localice la posición del poste. ','A')
```

```
IF ($button = 3)
```

```
    REMOVENUS ( )
```

```
    RETURN ( )
```

```
ENDIF
```

```
WRITEDBREC ('A', '')
```

2. Opción ' B ' :

```
SETSYMNBR ( 30)
```

```
SETSNAP ('L')
```

```
PLCSYMBOL ('S ',' ','IA')
```

```
FLDGET ('',F_NAME)
```

```
WRITEDBREC ('B', '')
```

3. Opción 'L':

Permite construir al usuario una facilidad lineal (PIPE) usando la instrucción GDRAW. Escribe la información al archivo .NGF usando una rutina que sucederá a pesar de todo.

1. Selecciona el primer punto de la tubería a un archivo no gráfico.
2. Selecciona el punto final de la tubería a un archivo no gráfico.

4. UTILITARIOS DE CABLECAD

CableCad presenta muchos utilitarios que le permiten una customización volviéndose de esta manera más funcional para el usuario, para aplicaciones de programación él más importantes es el utilitario GEN porque permite manipular los menús, de esta manera Ud. crea sus UDC's y puede presentarlas en un menú de pantalla.

4.1. UTILITARIO GEN

En esta utilidad se tiene archivos con las siguientes descripciones *.OB que son los archivos del prompt , los *.SB que son los archivos de mensajes, *.MB son archivos binarios que controlan los menús y mensajes en la pantalla.

Para hacer los cambios necesarios se utiliza la utilidad GEN , pero si se desea hacer cambios mayores lo más conveniente es editarlo en una versión ASCII .

MODIFICACIÓN DE MENÚS , PROMPTS , Y MENSAJES

Desde el grupo de CABLECAD se selecciona GEN , la pantalla del GEN aparece , con un listado de los archivos *.OB , *.SB , *.MB , se puede escoger cualquiera de los archivos , cuando se escoge aparece en la pantalla el menú o mensaje correspondiente.

Esta utilidad utiliza su propio editor en el cual se puede añadir , borrar o modificar cualquier ítem seleccionado , se puede poner la posición o cambiar el color

Para usar el editor de la utilidad GEN se necesita conocer las siguientes teclas funcionales:

<TAB>	La llave TAB le permite moverse al siguiente ítem
<SHIFT>+<TAB>	Esta combinación de llaves le permite ir al ítem previo
<HOME>	Esta llave pone el cursor al principio del ítem
<CTRL>+<HOME>	Esta llave te coloca en la primera opción en el archivo.
<END>	Esta llave pone el cursor al final del ítem
<CTRL>+<END>	Esta llave te coloca en la última opción en el archivo
<ALT>+A	Esta combinación le permite adicionar una letra o palabra al final del ítem
<ALT>+D	Esta combinación borra el ítem de la opción
<ALT>+F	Esta combinación remueve la opción
<ALT>+G	Esta combinación le permite presentar el número del ítem especificado
<ALT>+H	Esta combinación presenta una ayuda
<ALT>+I	Esta combinación es para insertar una letra
<ALT>+N	Esta combinación permite insertar un nuevo ítem en la opción
<ALT>+O	Esta combinación permite modificar el campo OPCODE en los archivos .OB
<ALT>+S	Esta combinación permite buscar a un string
<F1>	Esta llave se usa para mover el menú a la derecha
<F1>+<SHIFT>	Esta llave se usa para mover el menú a la izquierda
<F2>	Esta llave permite mover el menú hacia abajo
<F2>+<SHIFT>	Esta llave permite mover el menú hacia arriba
<F3>	Esta llave se usa para incrementar en un espacio a la derecha del menú
<F3>+<SHIFT>	Esta combinación se usa para decrementar un espacio a la derecha del menú
<F4>	Esta llave es para incrementar un espacio a lo largo del menú
<F4>+<SHIFT>	Esta combinación es para decrementar un espacio a lo largo del menú
<F5>	Esta llave permite cambiar el color del fondo del menú
<F6>	Esta llave permite cambiar el color de las letras del menú
<F7>	Esta llave cambia el resaltador en el menú
<F8>	Esta llave cambia el resaltador de una letra

5. DESARROLLO DEL SISTEMA

El sistema a desarrollar consta de una base de datos gráfica y no gráfica aplicada a un sistema de Distribución Eléctrica llamado "CURSO".

CURSO:

Es un modulo de aplicaciones básicas pero muy poderosas, que trata de realizar tareas muy practicas sobre aplicaciones de Distribución Eléctrica.

Para ello se crea una serie de UDC'S las cuales se describen a continuación.

Para la elaboración de los siguientes programas existió la necesidad de crear símbolos y registros.

Los símbolos fueron creados por medio del programa SYMGEN, mientras que para los registros se utilizo el SCRIPT, todo esto bajo una base de datos denominada engen_40.ftm \ engen_40.idx que se encuentran ubicados en el directorio d:\cablecad

Curso se encuentra ubicado en el menú ENGEN en la ultima posición, la función de curso es llamar ha una UDC denominada curso.udc, la cual se encarga de cargar el menú principal.

5.1. MENU "CURSO"

Se encarga de llamar a curso.udc y presentar el menu "CURSO".

El menú CURSO presenta siete opciones especificas.

Cada opción llama a la UDC correspondiente ubicada en D:\ENGEN\CURSO

Código:

```
REMOVEMENUS()
TEXTCOPY(@TEMP,'lengen\')
TEXTCAT(@TEMP,'IDSP\MAINMENU')
SETMENUMBR(@TEMP,'12')
TEXTCOPY(@TEMP,'lengen\')
TEXTCAT(@TEMP,'CURSO')
SETDIR(@TEMP)
```

CURSO LINEA TRANSFOR ABONADO CONSULTA MODIFICAR BORRAR BASEPLAN
--

Descripción de las opciones:

LINEA :

Es una UDC que se encarga de llamar a un menú compuesto por dos opciones:



Código:

```
Global (@MAIN_DIR,'TEXT')
SETMODE('N')
TEXTCOPY(@TEMP,@MAIN_DIR)
TEXTCAT(@TEMP,'\DSP\MAINMENU.OB)
SETMENUMBR(@TEMP,'14')
```

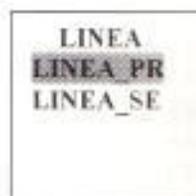
LINEA PR

Crea una línea inteligente primaria en cual se puede ingresar datos.

No Retorna ningún valor

Código:

```
SETDBREC(CABLE1)
SETSYMNBR(176)
SETSYMSIZE(2)
/*CABLEDRAW(29)*/
GDRAW()
FLDGET('INGRESE CAMPO1:',CAMPO1)
FLDGET('INGRESE CAMPO2:',CAMPO2)
WRITEDBREC('L',")
```



REMOVEMENUS()

RETURN()

LINEA SC

Crea una línea inteligente secundaria en cual se puede ingresar datos.

No Retorna ningún valor

Código:

SETDBREC(CABLE2)

SETSYMNBR(176)

SETSYMSIZE(2)

SETGCOLOR(RED)

GDRAW()

/*CABLEDRAW(29)*/

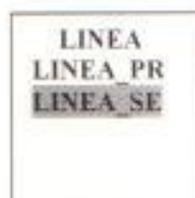
FLDGET('INGRESE CAMPO1 :',CAMPO1)

FLDGET('INGRESE CAMPO2 :',CAMPO2)

WRITEDBREC('L',")

REMOVEMENUS()

RETURN()



TRANSFOR

Esta UDC se encarga de la creación de un transformador.

No Retorna ningún valor

Código:

SETDBREC(TRANSFO)

SETSNAP('V')

SETSYMNBR(39)

SETSYMSIZE(8.0)

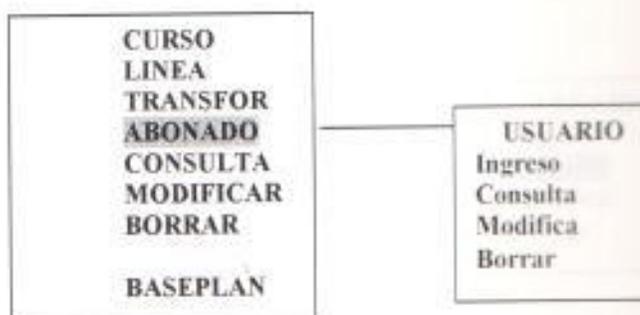
PLCSYMBOL('B','INDIQUE LOCALIZACION :','O')

WRITEDBREC('C',")



ABONADO

Es una UDC que se encarga de llamar a un menú compuesto por tres opciones:



Código:

```
Global (@MAIN_DIR,'TEXT')
SETMODE('N')
TEXTCOPY(@TEMP,@MAIN_DIR)
TEXTCAT(@TEMP,'DSP\MAINMENU.OB)
SETMENUMBR(@TEMP,'13')
```

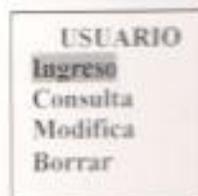
INGRESO

Esta UDC se encarga del ingreso de datos del abonado.

No Retorna ningún valor

Código:

```
SETDBREC(NUSUARIO)
/*SETSNAP('V')*/
SETSYMNBR(175)
SETSYMSIZE(50.0)
SETGCOLOR(GREEN)
FLDGET('INGRESE NOMBRE :',NOMBRE)
FLDGET('INGRESE DIRECCION:',DIRECCION)
FLDGET('INGRESE TELEFONO:',TELEFONO)
PLCSYMBOL('B','INDIQUE LOCALIZACION :','O')
WRITEDBREC('C','')
RETURN()
```



CONSULTA

Esta UDC se encarga de la consulta de datos del abonado.
No Retorna ningún valor

Código:

```
SELECT('SELECCIONE ELEMENTO A CONSULTAR')
IF ($BUTTON = '3')
    REMOVEMENUS()
    RETURN()
ENDIF
SBROWSE_VIEWONLY()
RETURN()
```

USUARIO
Ingreso
Consulta
Modifica
Borrar

MODIFICA

Esta UDC se encarga de modificar los datos del abonado.
No Retorna ningún valor

Código:

```
SELECT('SELECCIONE ELEMENTO A MODIFICAR')
IF ($BUTTON = '3')
    REMOVEMENUS()
    RETURN()
ENDIF
SBROWSE()
RETURN()
```

USUARIO
Ingreso
Consulta
Modifica
Borrar

BORRAR

Esta UDC se encarga de borrar los datos del abonado.
No Retorna ningún valor

Código:

```
DELETEITEM(0)
```

USUARIO
Ingreso
Consulta
Modifica
Borrar

BORRAR

Esta UDC se encarga de borrar.

No Retorna ningún valor

CURSO LINEA TRANSFOR ABONADO BORRAR CONSULTA MODIFICAR BASEPLAN

Código:

```
DELETEITEM(0)
```

CONSULTA

Esta UDC se encarga de la consulta de datos del abonado.

No Retorna ningún valor

CURSO LINEA TRANSFOR ABONADO BORRAR CONSULTA MODIFICA BASEPLAN
--

Código:

```
SELECT('SELECCIONE ELEMENTO A CONSULTAR')  
IF ($BUTTON = '3')  
    REMOVEMENUS()  
    RETURN()  
ENDIF  
SBROWSE_VIEWONLY()  
RETURN()
```

MODIFICA

Esta UDC se encarga de modificar los datos del abonado.
No Retorna ningún valor



Código:

```
SELECT('SELECCIONE ELEMENTO A MODIFICAR')
IF ($BUTTON = '3')
    REMOVEMENUS()
    RETURN()
ENDIF
SBROWSE()
RETURN()
```

BASEPLAN

Es una función propia de CableCad que posee aplicaciones propias de dibujo.

5.2. DEFINICION DE REGISTROS

A continuación se muestra la estructura de los diferentes registros utilizados

Nusuario			
	Nombre	Dirección	Teléfono
Record Numb	884	884	884
Graphic Numb	1	1	1
Record Name	Nusuario	Nusuario	Nusuario
Item Type	Symbol	Symbol	Symbol
Colour	2	1	1
Style	1	1	1
Weight	1	1	1
Level	1	1	1
Feature Code	1	1	1
Txt Symbol Size	1	1	1
Reserved Field	0	0	0
Descripción	Nombre	Dirección	Teléfono
Field Length	30	30	10
Data Type	Alphanumeric	Alphanumeric	Alphanumeric
Decimal	0	0	0
Data Edit			
Visible	Visible	Visible	Visible

CABLE1

	Campo1	Campo2
Record Numb	885	885
Graphic Numb	1	1
Record Name	CABLE1	CABLE1
Item Type	Symbol	Symbol
Colour	1	1
Style	1	1
Weight	1	1
Level	1	1
Feature Code	1	1
Txt Symbol Size	1	1
Reserved Field	0	0
Descripcion	Campo1	Campo2
Field Length	30	30
Data Type	Alphanumeric	Alphanumeric
Decimal	0	0
Data Edit		
Visibilite	Visible	Visible

CABLE2

	Campo1	Campo2
Record Numb	886	886
Graphic Numb	1	1
Record Name	CABLE2	CABLE2
Item Type	Symbol	Symbol
Colour	1	1
Style	1	1
Weight	1	1
Level	1	1
Feature Code	1	1
Txt Symbol Size	1	1
Reserved Field	0	0
Descripcion	Campo1	Campo2
Field Length	30	30
Data Type	Alphanumeric	Alphanumeric
Decimal	0	0
Data Edit		
Visibilite	Visible	Visible

TRANSFO

	Record Numb	Campo1
		887
	Graphic Numb	1
	Record Name	CABLE2
	Item Type	Symbol
	Colour	1
	Style	1
	Weight	1
	Level	1
	Feature Code	1
	Txt Symbol Size	1
	Reserved Field	0
	Descripcion	Campo1
	Field Length	30
	Data Type	Alphanu

CONCLUSIONES

Cuando se intenta comparar los varios enfoques al diseño de GIS, el grado de apertura debe ser reconocido como una cuestión pero al mismo tiempo un sentido de perspectiva es importante, por ejemplo podría decirse que en algún lenguaje de programación de nivel básico es un sistema lo suficientemente abierto para construir un GIS y por lo mismo, algún sistema que provee una interface de programación podría ser descrito como un sistema abierto. Mientras no haya nada incierto acerca de esta afirmación, el aspecto económico juega una parte importante y exige que encontremos un paradigma más eficiente y poderoso. Este documento ha descrito dos componentes de dicho paradigma: una interface de base de datos virtual y un lenguaje interactivo orientado al objeto front-end.

Una función llena (completa) fue diseñada para resolver problemas de apariencia para utilidades y telecomunicaciones de compañías, ahora y en el futuro.

Se trata de reducir el tiempo de implementación dando el más rápido uso operacional y minimizando el tiempo requerido para realizar un regreso en su inversión.

Se trata de conservar su inversión en tecnología, aprovechando la industria y usando un sistema de arquitectura abierto.

Si esta creando diseños o datos, analizando un problema o preguntando una base de datos. GIS le ofrece el más efectivo medio para llevar a cabo su negocio.

CableCad se presenta mas como AM\FM con ciertas características GIS, como sistema presenta una interface agradable al usuario, presentando menús y submenús, los cuales pueden ser manipulados muy fácilmente usando el mouse o aplicando las teclas direccionales, estos menús se ven sustentados

por herramientas utilitarias propias de CableCad con una serie de aplicaciones practicas, dando gran facilidad al usuario.

Permite realizar mapas de manera continua sin la necesidad de seccionarlos en partes permitiendo de esta manera tener una visión más amplia del gráfico. Tiene una escala ajustable permitiendo realizar gráficos adaptados a ella.

Posee conectividad total esto significa que cada elemento del gráfico puede estar conectado entre sí, siempre que los elementos sean compatibles, además de esto tiene la facilidad de generar reportes.

Al tener dos Bases de Datos Gráfica y no Gráfica permite una gran asociación de elementos, haciendo más fácil la tarea de mantenimiento de datos, permite realizar una manipulación de datos y generarlos.

Por poseer un macrolenguaje presenta una gran cantidad de funciones ya definidas por CableCad, se presenta como un lenguaje modular, esto da facilidad a la creación de nuevos programas y esta a su vez asociarlos a menús.

CableCad por presentarse en un sistema de arquitectura abierta permite una Base de Datos compartida por un numero de usuarios, unidades de programas independientes con facilidades de acceso al código fuente.

Todo esto envuelto en un entorno que permite la digitalización y con ello la captura de datos de manera más efectiva.

En el presente trabajo se demuestra que por medio de la utilización del lenguaje de programación que aplicado a las bases de datos gráfica y no gráfica es posible desarrollar Sistemas de Ingeniería que satisfagan los requerimientos de los Sistemas de Distribución de Empresas Eléctricas.

Recomendación

Se ha realizado los programas de "CURSO" , los cuales deben de ser alimentados con los datos técnicos apropiados para el diseño, análisis, y operación de los Sistemas de Distribución .

Bibliografía

1.-Informacion tomada de Internet :

Direcciones:

WWW.altavista

WWW.enghouse.com

2.- Apuntes facilitados por el Ing Juan Saavedra Mera

3.- Manuales Enghouse de la aplicación CABLECAD