

T
005.76
MOR
C2



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

SISTEMA DE INFORMACION GEOGRAFICO
APLICADO AL TURISMO (SIT)
"PLAN PILOTO PARA LA PROVINCIA DEL GUAYAS"



TRABAJO DE GRADUACION SISTEMAS AM/FM GIS

6/20/01
Previo a la obtención del título de:
INGENIERO EN COMPUTACION



D-22899

PRESENTADO POR:



SYLVIA MOREIRA NUÑEZ
VERONICA MURGA JARAMILLO
SILVIA ZEA VIDAL

Guayaquil - Ecuador
2001



AGRADECIMIENTO

Al Ing. Oscar Jaramillo, Asesor del Proyecto y al Ing. Javier Urquiza, Director del Tópico por su ayuda y colaboración para la realización de este proyecto




DEDICATORIA

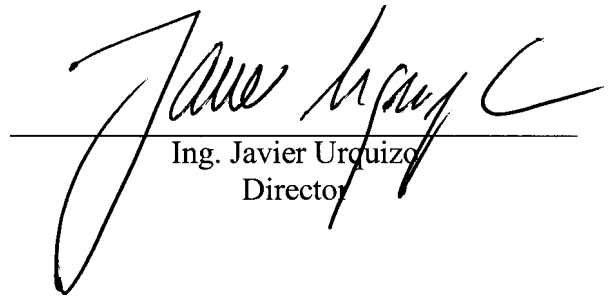
A NUESTROS
Padres quienes
siempre han tenido
palabras de aliento y
de quienes hemos
tenido un
incondicional apoyo
desde el inicio de
nuestros estudios. A
familiares que han
apoyado con su ayuda
y comprensión.



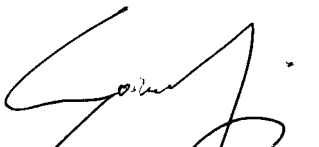
TRIBUNAL DE GRADUACION




Ing. Carlos Monsalve
Presidente



Ing. Javier Urquiza
Director



Ing. Guido Caicedo
Miembro Principal

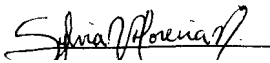


Ing. Mónica Villavicencio
Miembro Principal

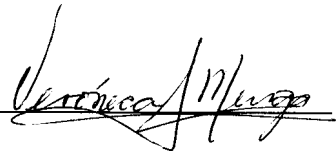
DECLARACION EXPRESA

“La responsabilidad del contenido de esta tesis nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

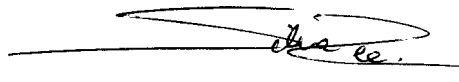
(Reglamento de Graduación de la ESPOL)



Sylvia Moreira Núñez



Verónica Murga Jaramillo



Silvia Zea Vidal



RESUMEN



El siguiente documento recopila información acerca de la concepción, diseño y desarrollo de un sistema de información geográfico para el turismo, dicho proyecto forma parte importante en el proceso de graduación, para él se han utilizado herramientas de software para el manejo de datos y la información geográfica. El desarrollo de este software ha sido realizado usando la información obtenida en la Subsecretaría de Turismo y las herramientas actuales de programación.

De entre los diferentes temas propuestos por el Director del Tópico, este nos ha sido asignado, y, estamos seguras que será implementado por otras Direcciones de Turismo y que no solamente será un Plan Piloto para la provincia del Guayas, este es un programa supremamente necesario para la entidad con la que se ha estado trabajando, pues marca un comienzo importante en la modernización de esta dependencia del estado. No está realizado con las herramientas usuales del Centro de Desarrollo de Proyectos debido a que la principal cualidad que es su bajo costo.

Este programa ha sido realizado con mucho esfuerzo pensando siempre satisfacer todas las necesidades tanto del usuario como de la dependencia para la cual se lo desarrolló, esperamos pues que cumpla con las expectativas de todos aquellos quienes de una forma u otra se han visto inmersos en este proyecto.



INDICE GENERAL

AGRADECIMIENTO	11
DEDICATORIA	111
TRIBUNAL DE GRADO	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
INDICE GENERAL	VII
INTRODUCCION	VIII

CAPITULO 1

Proyecto Sistema de Información Turístico

Estudios sobre el proyecto

1.1 Antecedentes.....	2
1.2 Objetivos del proyecto	3
1.3 Análisis del proyecto	4
1.4 Estrategias de solución	5
1.4.1 Levantamiento de la información turística	6
1.4.2 Levantamiento de la información geográfica	7

CAPITULO 11

Sistema de Información Turístico Nacional e Internacional

Esfuerzos alcanzados en el desarrollo de Sistemas de Información Turístico

2.1 SIT (Sistema de Información Turístico) en la actualidad.....	9
2.2 Búsqueda de soluciones de Sistemas de Información Turístico.....	10
2.3 Prospectos para el futuro	12

CAPITULO III

Estrategias y diseño del Proyecto Sistema de Información Turístico

Especificaciones del proyecto

3.1 Descripción general del proyecto	15
3.2 Estrategias de desarrollo	17
3.2.1 Levantamiento de la información	18

3.3	Diseño y Desarrollo del sistema Edén	19
3.3.1	Información de los datos de las plantas turísticas	20
3.3.2	Información Geográfica de la planta turística	28
3.3.3	Mantenimiento de Datos	29
3.3.4	Reportes	30

CAPITULO IV

Implementación, Pruebas y Resultados

Sistema de Información Turístico en la Subsecretaría de Turismo

4.1	Características del sistema Edén.....	32
4.2	Resultados obtenidos	33

CONCLUSIONES Y RECOMENDACIONES35
---	------------

ANEXOS

Anexo A.	Modelo del Sistema de Información Turístico (SIT) Eden
Anexo B.	Manual del Usuario del Sistema de Información Turístico (SIT) Eden
Anexo C.	Manual de Instalación del Sistema de Información Turístico (SIT) Eden
Anexo D.	Enlaces del Mapa a la base de datos del Sistema de Información Turístico (SIT) Eden
Anexo E.	Manual de Georeferenciación y Digitalización
Anexo F.	Tablas de la base de datos Edén
Anexo G.	Modelo Entidad Relación
Anexo H.	Diccionario de Datos
Anexo 1.	Código fuente

BIBLIOGRAFIA

INTRODUCCION

El Centro de Desarrollo de Proyectos de la Escuela Superior Politécnica del Litoral a través de los años ha trabajado en varias propuestas de alta tecnología a bajo costo y no ha sido la excepción este Tópico Especial de Graduación, en el cual se ha logrado dar solución al problema del catastro para los establecimientos turísticos de la provincia del Guayas.

Este documento contiene toda la información de un proyecto realizado para el Ministerio de Turismo. El capítulo 1 hace una breve reseña de los estudios previos sobre este tema y explica las mejoras, objetivos y estrategias del proyecto actual.

En el capítulo II se hace una referencia a los sistemas nacionales e internacionales que son utilizados en la actualidad. Además se mencionan algunas herramientas para sistemas de información geográficos (GIS).

En el capítulo III se habla de las partes del proyecto y su estrategia de desarrollo. Explica las herramientas que se van a utilizar y lo que se hará con toda la información recopilada tanto turística como geográfica.

En el capítulo IV se verán las características del sistema y su funcionalidad a beneficio del Ministerio de Turismo.

Por último, las recomendaciones, conclusiones del desarrollo del proyecto y los anexos. En los anexos se incluyen los manuales necesarios para entender el manejo y la utilidad de las herramientas necesarias para el desarrollo del sistema, información sobre la base de datos del sistema Edén así como el manual del usuario del mismo.

Capítulo I

I. PROYECTO SISTEMA DE INFORMACIÓN TURÍSTICO

ESTUDIOS SOBRE EL PROYECTO

1.1 Antecedentes

Años atrás ya se ha empezado a desarrollar un sistema de información turístico y ha estado en expectativas de mejoras para que su funcionamiento sea el más óptimo. Los estudios realizados en tesis anteriores basaron su proyecto en la utilización de una herramienta llamada “address match” que localizaba la dirección que deseaba el usuario. Además en ese proyecto se desarrolló una aplicación que permitía, adicional a la ubicación geográfica de un lugar turístico, ver la foto respectiva de ese lugar. Se consideraba como lugar turístico las iglesias, parques y demás lugares recreacionales.

El proyecto anterior sólo daba información de la parte céntrica de la ciudad de Guayaquil porque son las que tienen nombres de calles, las ciudadelas no podían localizarse en el mapa porque su dirección se conoce por manzana y villa.

El proyecto actual que nosotras presentamos tiene un enfoque mucho más grande y utiliza herramientas más sofisticadas para su desarrollo. Además recopila la información turística de toda la provincia del Guayas y su implementación es a beneficio de la Subsecretaría de Turismo.

Por lo tanto, la misión de este proyecto es dotar a la Subsecretaría de Turismo de una herramienta eficaz de control y planificación de sitios turísticos, sean estos lugares de diversión, comidas y bebidas, alojamiento, transporte, etc. , es decir, toda la información que una institución como ésta pueda proveer públicamente a quien lo necesite.

1.2 Objetivos del proyecto

El proyecto tiene como finalidad:

- Contar con un sistema de información geográfico de todos los establecimientos que controla la Subsecretaría de Turismo.
- Poder ver la información de un establecimiento junto con su ubicación geográfica y su respectiva foto.
- Satisfacer los requerimientos de la Subsecretaría de Turismo para un eficaz control de los establecimientos turísticos.
- Ofrecer un sistema de información turístico a bajo costo.

1.3 Análisis del proyecto

Con el conocimiento de los estudios realizados años anteriores, se ha buscado mejoras para la aplicación de este proyecto en el campo real, que sería en este caso, en el Ministerio de Turismo.

El sistema de información turístico debe ser un sistema abierto, es decir, que permita al usuario poner su propia terminología sin tener necesidad de llamar a los desarrolladores del sistema para modificarlo a su gusto. Por ejemplo, el usuario pueda ingresar las diferentes formas de pago, los motivos de pago, las actividades turísticas, las categorías, la terminología como activo, inactivo, retirado, clausurado, etc., imprimir un reporte de los datos que desea, hacer una consulta por cualquier dato del establecimiento, ya sea el nombre, registro, dirección, fecha de clausura, fecha de inicio de operaciones, por actividad, categoría, etc.

Además el sistema debe permitir ver los datos generales, la ubicación geográfica y la foto de un mismo establecimiento a la vez.

Para que el sistema sea eficaz y pueda ser utilizado por el Ministerio de Turismo, debe:

- Tener los mismos datos que ellos manejan por cada establecimiento.

- Imprimir todos los reportes necesarios, en especial el catastro turístico.
- Llevar un control de pagos que los establecimientos deben realizar.
- Tener la ubicación geográfica de cada establecimiento de la provincia del Guayas.
- Permitir la búsqueda de cualquier establecimiento por cualquier dato (registro, nombre, dirección, fecha inicio de operaciones, fecha clausura, categoría, etc.).

Una de las metas en el diseño y codificación de este sistema, es lograr que cumpla todos los requerimientos posibles y futuros del usuario, razón por la cual, la flexibilidad en el diseño es indispensable.

1.4 Estrategias de solución

El proyecto se dividió básicamente en dos partes:

- Recopilación de datos: información escrita, geográfica y foto
- Desarrollo del software.

Para el desarrollo del software se realizaron entrevistas con la Subsecretaría de Turismo para conocer sus requerimientos en el manejo de los datos y el formato de impresión de los reportes.

A continuación se detalla la recopilación de datos.

1.4.1 Levantamiento de la información turística

Para la recopilación de información escrita se utilizó el catastro de diciembre de 1.999 que tiene la Subsecretaría de Turismo. Luego se realizó un censo para la actualización de los datos.

Para el censo se recogió en formularios preparados la información descriptiva de cada Planta Turística.

Toda la información que se recopiló en el censo se utilizó para el diseño de la base de datos.

Se utilizaron cámaras digitales para tomar las fotos a los establecimientos.

1.4.2 Levantamiento de la información geográfica

Para la información geográfica se obtuvo la cartografía de las cuadrículas que conforman toda la provincia del Guayas en donde la

fuelle fue el Instituto Geográfico Militar. El proceso de digitalización de la información geográfica se describe en el Anexo E.

Para la información geográfica también se consideró importante ver el área en que está ubicado el establecimiento como la altura, extensión, ríos parajes, etc.

Capítulo II

11. Sistema de Información Turístico Nacional e Internacional

Esfuerzos alcanzados en el desarrollo de Sistemas de Información Turístico

2.1 SIT (Sistema de Información Turístico) en la actualidad

En la Subsecretaría de Turismo de la provincia del Guayas, antes de la implementación del sistema Edén, no contaban con un sistema de Información Geográfico Turístico. En las oficinas de dicha entidad, tenían un programa que sólo permitía almacenar, sacar reportes y actualizar la información de las plantas turísticas. Pero era un programa muy limitado, pues no permitía filtrar la información como deseaba el usuario. Esto hizo que el programa sea desplazado y olvidado para el control de las plantas turísticas.

Lamentablemente, el gobierno no se ha preocupado por mejorar el Turismo, por mejorar la imagen de nuestro país a nivel internacional. No hay un incentivo a los turistas por visitar nuestro país, a pesar de que el Turismo es un ingreso beneficioso para todos los ecuatorianos.

El Ecuador no tiene páginas en el Web que muestren sus lugares turísticos. Mientras que en otros países tales como Uruguay, el Ministerio de Turismo se ha preocupado por mostrar por medio del Internet planos de su país, fotos de

cada ciudad, mapas de las rutas, etc. También otros países latinoamericanos y europeos tienen un menú variado que incentiva el turismo, me refiero, a conciertos, festivales, espectáculos, planos de la ciudad, gastronomía, restaurantes, hoteles, paquetes turísticos, ferias, convenciones, fotos de la ciudad, rutas, teléfonos de interés, situación, clima, museos, mercados, monumentos, compras, playas, etc.

2.2 Búsqueda de soluciones de Sistema de Información Turístico

Para desarrollar Sistemas de Información Turístico útiles para la comunidad se han realizado estudios e investigaciones con el objeto de mejorar las herramientas GIS (Geographic Information System). GIS es un sistema automatizado compuesto de hardware, software y datos que permiten crear, almacenar, mostrar y analizar datos especiales y atributos relacionados. En algunas universidades de Estados Unidos existen equipos de desarrolladores de programas que utilizan herramientas GIS para el desarrollo de sistemas que manejan datos geográficos.

Hay algunas herramientas GIS, adicionales a la que utilizamos en este proyecto, que permiten tener buenos resultados en la búsqueda de un sistema de información geográfico. La tecnología GIS ha madurado a tal punto que

hay expectativas muy altas. Es un potencial para nuevos paradigmas como el producir cambios en la manera de pensar de la gente de una información geográfica.

Existe una gran variedad de herramientas para diseñar sistemas de información geográfico turístico, a continuación se nombra algunas de ellas:

- **GRASS GIS** (Geographic Resources Analysis Support System) es un recurso de GIS con raster, vector topológico, procesamiento de imagen y producción de gráficos funcionales que operan en varias plataformas con una interfase gráfica. Actualmente es usado por academias, áreas comerciales de todo el mundo, así como agencias gubernamentales incluyendo NASA, The National **Park** Service y muchas compañías consultoras de Medio Ambiente.
- **FMaps** provee funciones GIS,
- **OGR** es una librería de C++ para leer y escribir en formato vector
- **MapServer**, está desarrollado en ambiente OpenSource para construir aplicaciones espaciales en Internet. Trabaja en casi todos los UNIX, Windows NT/98/95. OpenSource, se dice de los programas que tienen libre redistribución que incluye su código fuente el cual puede ser modificado.
- **OSSIM**, Open Source Software Image Map, equipos de expertos de disciplinas como Ciencias de Información Geográfica y Cartografía

desarrollan códigos que el público puede copiarlos y compilarlos. Estos código están hechos en lenguaje C++

- **WinGIS 2000** version **3.4.4** tiene muchas funcionalidades para trabajar con mapas.
- **OpenMap** es un visor geo espacial en lenguaje Java.

También existen muchas compañías dedicadas al desarrollo de sistemas geográficos tales como la compañía canadiense GeoSolutions, la corporación GeoSolv, GDMS (Geographic Data and Managment Solutions), GeoStrategies de Cambrigde, SIGSA (Sistemas de Información Geográfico S.A.) de México, entre otros.

Casi todas las formas de uso geográfico pueden ser automatizadas, pueden ser vistos los mapas, atlas, la información geográfica puede ser analizada La información geográfica se hace mucho más poderosa en un ambiente digital.

2.3 Prospectos para el futuro

El alcance logrado con herramientas GIS para desarrollo de sistemas de información geográfico permite tener una visión de los beneficios que se pueden obtener en el futuro: Una geografía automatizada, podemos usar datos espaciales para propósitos específicos o para desarrollar teorías generales,

GIS es un **gran** aporte para el desarrollo de la ciencia de información espacial, la cual podría incluir: colección de datos, cartografía, análisis espacial, GIS es una herramienta científica en la explicación y comprensión del mundo que nos rodea.

Capítulo III

111. ESTRATEGIAS Y DISEÑO DEL PROYECTO SISTEMA DE INFORMACIÓN TURISTICO

ESPECIFICACIONES DEL PROYECTO

3.1 Descripción general del proyecto

Por ser un paso enorme hacia la modernización de la Subsecretaría de Turismo se requirió que el diseño presentara mucha flexibilidad para prever cambios en el futuro, pero se debe recalcar que la calidad y funcionalidad son características indudables del diseño de este software por lo que en breve la estabilidad del mismo no se verá afectada.

El sistema tendrá un user y password para que sólo tengan acceso a él las personas autorizadas.

El proyecto se desarrolló en ambiente Windows, por lo tanto tiene las mismas características de Windows como son listas desplegables, cajas de texto, botones, etc.

El sistema permitirá ingresar los datos que tienen los formularios de planta turística. Estos formularios los llena manualmente el supervisor de la Subsecretaría de Turismo.

Permitirá registrar y llevar a cabo un control de los establecimientos que hacen el pago anual de permiso de funcionamiento y los demás pagos requeridos.

El sistema debe manejar los datos de tal forma que se pueda modificar, agregar, eliminar y consultar.

El sistema debe permitir ver y obtener un informe de establecimientos ya sean activos, clausurados, no turísticos, retirados, etc.

Permitirá imprimir la información catastral, información de pagos al Ministerio y demás reportes y listados requeridos por la Subsecretaría de Turismo.

Al hacer una consulta, podrá ver los datos, la ubicación geográfica y la foto de la planta turística (establecimiento). Además pueda sacar la información geográfica del establecimiento para imprimirla.

3.2 Estrategias de desarrollo

Para el desarrollo del proyecto se buscó herramientas que cumplan con los objetivos especificados. Por lo tanto, se llegó a la conclusión de utilizar las siguientes herramientas:

Visual Basic 6.0 para manejo y control de toda la información turística.

La aplicación Dr. DWG View Professional 5.0 que es un visor del mapa (es un ocx para Visual Basic).

Microsoft Access para almacenar la base de datos.

Las dos últimas herramientas son las que el usuario final necesitaría tener en su computador para utilizar el sistema.

Sin embargo, nosotras para desarrollar la parte de información geográfica requerimos MicroStation para hacer los enlaces entre la información geográfica y los datos generales del establecimiento. Esta herramienta no necesita tener el usuario en su computador porque en su lugar es la aplicación Dr. DWG View Professional.

El Sistema de Información de Turístico “Edén”, fue desarrollado bajo la plataforma Visual Basic 6.0, usa un ODBC como herramienta para realizar la conexión entre el software de graficación geográfica “MicroStation” y la base

de datos. Por lo tanto si a futuro se desea migrar la base de datos es oportuno recalcar que el uso de ODBC nos permite el acceso a diversas Bases de Datos; así no nos veremos limitados al uso de Access como base de datos definitiva.

En base a los requerimientos y los formularios de planta turística que archiva la Subsecretaría de Turismo, se diseñó la base de datos cuyo diseño y diagrama entidad relación está en el Anexo G.

3.2.1 Levantamiento de la información

La información recopilada en el censo fue ingresada en una base de datos cuyo diseño se explica en el Anexo G.

Luego de haber digitalizado los mapas que se explican en el Anexo E, se ubicaron los puntos turísticos de la provincia del Guayas, colocando el nombre y código del establecimiento.

Se ingresó al sistema las fotos tomadas a los establecimientos.

Se realizó el enlace entre los mapas digitalizados y la base de datos turística.

3.3 Diseño y desarrollo del Sistema Edén

Luego de haber hecho un análisis de los objetivos de este proyecto se escogió las herramientas a trabajar que son:

Visual Basic, para desarrollo del sistema Edén

MicroStation para digitalizar los mapas

Access para almacenar la base de datos

Luego de las entrevistas con los señores de la Subsecretaría de Turismo, se hizo un análisis para el diseño de las tablas del sistema. Estas tablas las podrá ver en el apéndice F.

El diagrama entidad-relación se detalla en el apéndice G

El sistema se dividió en cuatro partes para facilitar su desarrollo y diseño:

1. Información de los datos de las plantas turísticas.
2. Información geográfica.
3. Mantenimiento de datos
4. Reportes

Además se organizaron en carpetas los archivos que utilizará el sistema. La carpeta principal es Edén, dentro de la cual están las subcarpetas:

Base

Grafico

Reportes

En la carpeta Edén estará el archivo ejecutable y el archivo.ini. Dentro de la carpeta Base estará sólo la base de datos Edén. En la carpeta gráficos estará todas las fotos en formato jpg y los mapas en formato dwg. En la carpeta Reportes estarán todos los reportes del sistema Edén.

3.3.1 Información de los datos de las plantas turísticas

La información que debe el usuario ingresar en el sistema:

- Dos formularios para la Planta Turística que llena el supervisor de la Subsecretaría de Turismo:
 - Un formulario específico para las agencias de Viaje
 - Un formulario para las demás actividades (alojamiento, comidas, etc)
- Se registra los establecimientos que hacen los pagos requeridos por la Subsecretaría de Turismo

Los dos formularios de planta turística serán ingresados en una misma ventana, por lo tanto, las agencias de viaje, alojamiento, comidas, recreaciones, etc., serán consideradas como tipo de actividades. Al ingresar una planta turística se podrá escoger de una lista desplegable el tipo de actividad.

Los tipos de actividad, están en la tabla tipo_actividad. a continuación se presenta el listado de algunos tipos de actividad:

ALOJAMIENTO
COMIDAS Y BEBIDAS
RECREACIONES Y REUNIONES
AGENCIAS DE VIAJE
CASINOS Y SALAS DE JUEGO
TRANSPORTES TURISTICOS
MATERIAL DE PROMOCION
INVESTIGACION Y CAPACITACION

Los formularios que llena el supervisor los dividimos en tres partes:

1. Datos Generales
2. Instalaciones
3. Servicios

Los datos generales del formulario que llena el supervisor son:

Registro No.

Folio

Nombre del establecimiento

Nombre del propietario

Representante legal

Dirección

Provincia

Cantón

Ciudad

sector

Tipo de actividad

Actividad

Fecha

Categoría

R.U.C.

Teléfono

Fax

Inicio de operaciones

Observaciones

Todos estos datos se guardan en la tabla grupo_turistico. Una vez ingresados en el sistema se podrá modificar los datos a excepción del registro, tipo de actividad y actividad por ser campos clave.

Las instalaciones que se registran en los formularios de las plantas turísticas son:

Para alojamiento

Tipo	habitaciones			No.	No.	Telef	Aire	Ascen	Insono	Frigo	tv
	CB	SB	BC	Cama	Plaza	Fax	Acond	sores	rizac	Bar	
	P	P									
Sencilla											
Matrim											
Doble											
Triple											

Para comidas y bebidas

Tipos	No. Mesas	No. Butacas	No. Plazas	No. Baño
Restaurante				
Cafetería				
Fuente de soda				
Bar				
Drive Inn				
Discoteca				

Los servicios adicionales que prestan las plantas turísticas son marcados en el formulario en el siguiente formato:

	SI	NO
PISCINA		
BAR		
SALONES DE USO		
BOTES		
RESTAURENTE		
CAFETERIA		
PELUQUERIA		
GIMNASIO		
HIDROMASAJES		
SALA DE SQUAT Y		
JACUZZI		
BUSINESS CENTER		
SALONES DE BANQUETES		
SAUNA		
TURCO		
CLUB		

La información de instalaciones y servicios que guardará el sistema está dividida en tablas: `tipo_instalacion`, `instalación`, `detalle_instalacion`, `servicios`, `servicio_instalacion`, `tipo_servicio`, `grupo_instalacion`. Siendo las tablas `servicio_instalacion` y `grupo_instalacion` las que guardan la información de cada planta turística. Las otras tablas son para escoger los servicios e instalaciones que tiene la planta turística que se está ingresando. A continuación se muestra los listados de cada tabla.

tipo_servicio

codigo	descripcion	estad
01	PISCINA	A
02	BAR	A
06	SALONES DE	A
22	BOTES	A
23	RESTAURENTE	A
24	CAFETERIA	A
25	PELUQUERIA	A
26	GIMNASIO	A

tipo_instalacion

codigo	descripcion	estado
01	DEPORTIVAS	A
02	RECREACIONALES	A
03	CONFERENCIAS	A
04	ENTRETENIMIENTO	A
05	COMEDORES Y	A
06	HABITACIONES	A

Instalación

codigo	codigo_tipo_inst	descripcion	estado
01	02	DISCOTECA	A
01	05	RESTAURANTE	A
01	06	SENCILLA	A
02	05	CAFETERIA	A
02	06	MATRIMONIAL	A
03	05	FUENTE DE SODA	A
03	06	DOBLE	A
04	05	BAR	A
04	06	TRIPLE	A
05	06	CUADRUPLE	A
06	06	SUITE	A
07	06	SUITE JUNIOR	A
08	06	SUITE PRESIDENCIAL	A
09	06	SUITE IMPERIAL	A
10	06	APARTAMENTOS	A
11	06	SUITE IMPERIAL DE LUJO	A

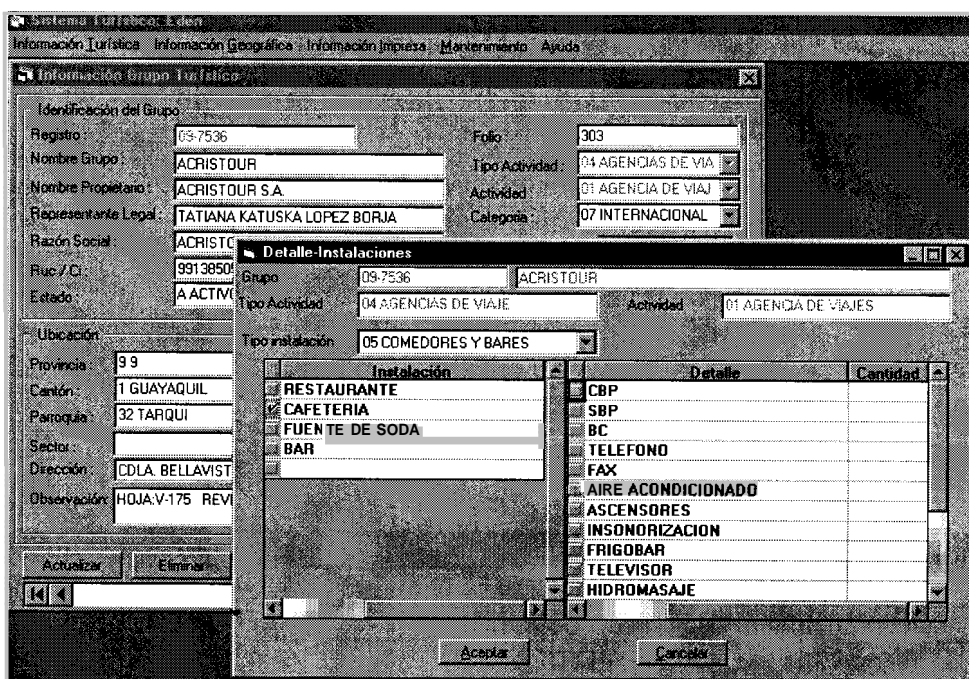
detalle-instalacion

codigo	descripcion	estado
01	CBP	A
02	SBP	A
03	BC	A
04	TELEFONO	A
05	FAX	A
06	AIRE ACONDICIONADO	A
07	ASCENSORES	A
08	INSONORIZACION	A
09	FRIGOBAR	A
10	TELEVISOR	A
11	HIDROMASAJE	A
12	SAUNA	A
13	MESAS	A
14	SILLAS	A
15	BUTACAS	A

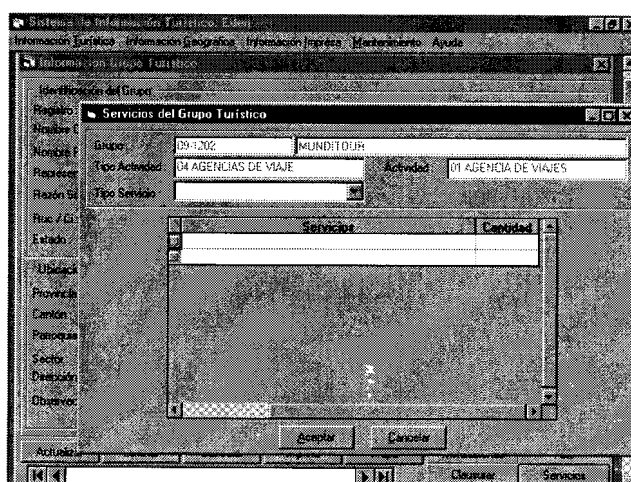
Servicio

codigo	codigo_tipo	descripcion	estado
01	22	CANOAS	A
02	22	KAYAK	A

La ventana diseñada para el ingreso de las instalaciones de la planta turística se muestra a continuación.



La ventana para el ingreso de los servicios de la planta turística:



Para los pagos que realiza la planta turística se diseñaron tablas de tal forma que el mismo usuario pueda ingresar en el sistema un listado de las formas de pago, tipo de pago y motivo de pago.

tipoqago

codigo	descripcion	signo	estado
NCR	CREDITO	1	A
NDB	DEBITO	-1	A

forma_pago

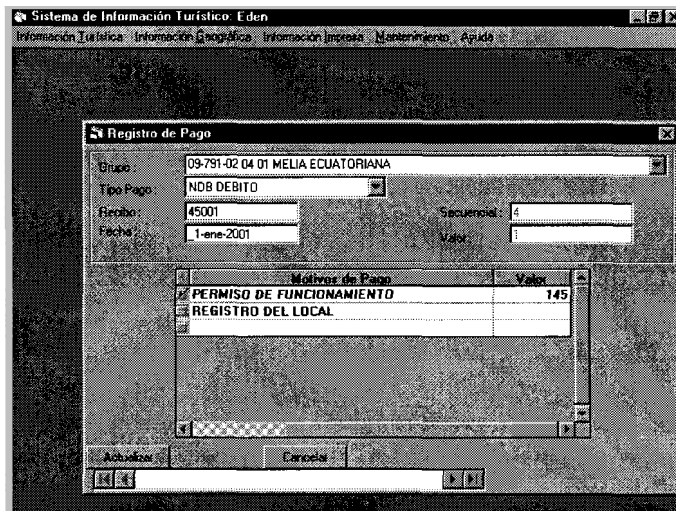
codigo	descripcion	estado
CHQ	CHEQUE	A
EFE	EFFECTIVO	A
PAP	PAPELETA DE	A

motivoqago

codigo	codigo_tipo_pago	descripcion	estado
FIS	NCR	CREDITO FISCAL	A
FUN	NDB	PERMISO DE FUNCIONAMIENTO	A
REG	NDB	REGISTRO DEL LOCAL	A

Además se diseñó dos tablas, cartera y cartera_detalle, para guardar los pagos que realiza la planta turística. Siendo cartera la cabecera y cartera_detalle guarda los diferentes motivos y forma de pago de la planta turística.

A continuación se muestra una ventana diseñada para ingresar un pago de un establecimiento.



3.3.2 Información Geográfica de la Planta Turística

Para la información geográfica, primero scaneamos el mapa del IGM obteniendo un archivo.cal (archivo raster). Este archivo lo georeferenciamos utilizando la herramienta Igeovec. Esto se explica en el anexo E.

Luego se procedió a la digitalización de las ciudades y cantones de la provincia del Guayas. Los pasos de la digitalización se explican en el mismo anexo E.

En la tabla grupo–turístico de la base de datos Edén se agregó el campo MSLINK que es el código que relacionará el mapa con los datos de la planta turística.

Luego de crear el código MSLINK se hizo el enlace entre el mapa digitalizado y la base de datos del sistema Edén. Para esto se utilizó la herramienta MicroStation. Esto se explica en el anexo D.

Después, se actualizó la tabla grupo–turístico generando el código del campo MAPID de dicha tabla. Este campo me permitirá enlazarme con el mapa que tiene la ubicación de la planta turística.

3.3.3 Mantenimiento de datos

Se consideró muy importante el área de mantenimiento de datos porque esta parte permite que el sistema sea flexible a los gustos del usuario. Mantenimiento de datos se refiere a tener listados de los diferentes datos que permiten tener un control de la información almacenada. Es decir, listado de actividades, tipo de actividades, servicios, tipos de servicios, formas de pago, motivos de pago, cantones, parroquias, etc.

3.3.4 Reportes

Esta parte es el diseño de los diferentes reportes solicitados por la Subsecretaría de Turismo. Para su diseño se utilizó la herramienta Cristal Reports que es una herramienta que trabaja con Visual Basic.

Capítulo IV

IV. IMPLEMENTACION, PRUEBAS Y RESULTADOS

FUNCIONES DEL SISTEMA DE INFORMACION TURISTICO

4.1 Características del sistema Edén

El sistema maneja los datos de acuerdo como son llevados en el Ministerio de Turismo.

Por otro lado, el funcionamiento es bastante sencillo tiene varias ventanas muy sencillas de manejar en las cuales se puede ingresar información, consultar, modificar o eliminar un sitio turístico. Además también si así se requiere se presenta la opción de ver la ubicación geográfica del lugar en un mapa de la ciudad al cual pertenezca, y si las exigencias lo solicitan, se puede ver una foto actualizada del sitio al que se está consultando.

Permite hacer un mantenimiento de todos los datos requeridos por las plantas turísticas, tales como actividades, categorías, instalaciones, servicios, formas de pago, motivos de pago, estado de grupo (activo, inactivo, no turístico, retirado, etc.)

Tiene los reportes requeridos por la Subsecretaría de Turismo tales como el informe catastral, que es el de mayor importancia, el informe de pagos, listados de actividades, categorías, instalaciones, servicios, empleados, etc. además tiene un informe parametrizado, es decir, que permite tener un informe según la información que el usuario desee filtrar.

4.2 Resultados obtenidos

El sistema de información turístico Edén es muy funcional porque cumple con los requerimientos de la Subsecretaría de Turismo.

El principal logro es la satisfacción de las partes fiscalizadoras, en la cual intervinieron miembros del Ministerio de Turismo en Quito, quienes luego de expresar su satisfacción con este Sistema de Información Turístico acordaron con el Centro de Proyectos que el plan piloto se ejecute a nivel nacional.

Es magnífico el alcance del proyecto porque cubre toda la provincia del Guayas y además permite también tener información de las demás provincias del país. Este sistema es útil para todas las provincias del Ecuador porque ayudará a controlar y mantener la información actualizada de todos los establecimientos de cada cantón, de cada parroquia.

Este sistema Edén es de gran utilidad para la Subsecretaría de Turismo porque los reportes que le ofrece el sistema son los que el Ministerio de Turismo en Quito solicitan.

La novedad de que se puede tener también una información geográfica conectada con los datos de las plantas turísticas también es de gran utilidad para el Ministerio de Turismo porque pueden observar en el mapa los lugares de mayor concentración de establecimientos como hoteles, comidas y bebidas, discotecas, casinos, bares, hostales, etc.

CONCLUSIONES Y RECOMENDACIONES

El Sistema de Información Turístico ha sido diseñado para ser ejecutado en diversas plataformas de la familia Windows, tales como: Windows 95, Windows 98, Windows 2000, Windows NT 4.0; de esta forma el ingreso, almacenamiento y consulta de la información se hará en forma muy similar al trabajo usualmente desarrollado sobre un PC común, esto hará que el trabajo para el usuario de “Eden” sea más eficiente y que pueda manejarlo sin complicaciones.

El sistema Edén siendo muy funcional para el Ministerio de Turismo pudiendo ser aplicado a nivel nacional es de bajo costo, pues el usuario (Ministerio de Turismo) sólo debe tener las licencias de Visual Basic y Microsoft Office que son de bajo precio en comparación con otras licencias como Oracle y MicroStation que también son herramientas para el desarrollo de este proyecto.

El sistema sí permite leer la información de un establecimiento, ver su ubicación geográfica y su respectiva foto. Además se puede modificar, buscar, eliminar e imprimir los datos de las plantas turísticas. También se puede sacar un impreso de la ubicación geográfica.

Satisface los requerimientos de la Subsecretaría de Turismo pues hace todos los reportes solicitados, tiene toda la información de la provincia del Guayas actualizada y su respectivo enlace con la información geográfica.

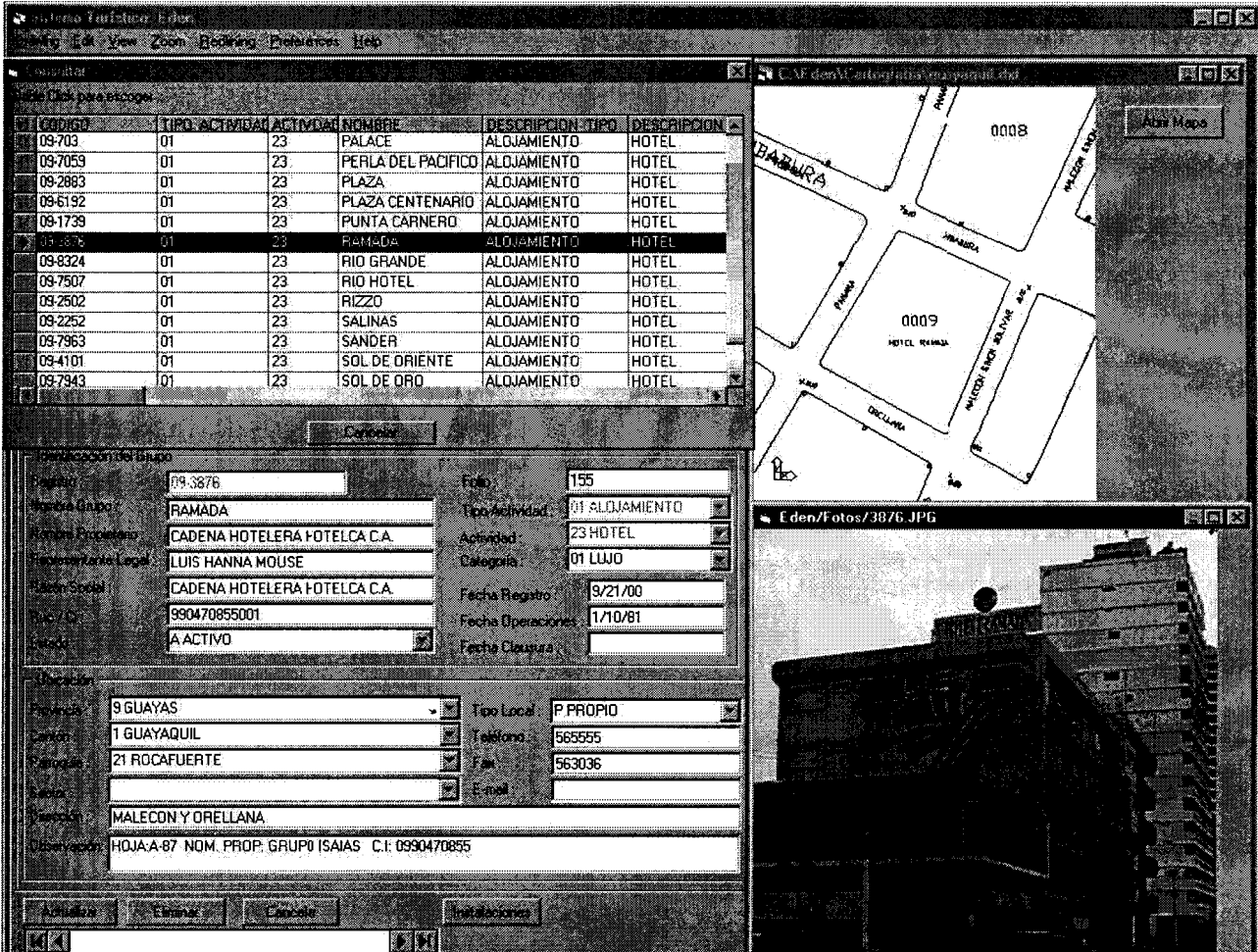
Este proyecto que comenzó como un plan piloto, y que ahora será llevado a cabo a nivel nacional, ha demostrado que debido a la excelente instrucción de los profesionales que prepara la ESPOL, existen muchas áreas en las cuáles se necesita de nuestro criterio como ingenieros. El Centro de Desarrollo de Proyectos debe continuar dando su aporte a la comunidad con la celebración de proyectos que presenten nuevos retos para los estudiantes que estamos a punto de culminar nuestras carreras, y, por supuesto está implícito el apoyo de los directivos de la universidad hacia este centro.

La Subsecretaría de Turismo de la Provincia del Guayas tiene el siguiente plan para el año 2001: Capacitación turística al personal del Ministerio, a los transportistas, capacitación en la calidad de servicios hoteleros, capacitación a los Guías y a las Agencias de Viajes. Marketing para Turismo interno y receptivo (impresión de afiches, mapas, planos, participación en ferias nacionales), inventario de atractivos turísticos y Turismo Sostenible, este último se refiere a un turismo permanente y actualizado con redes de información turística, centro de turismo comunitario, etc. En este último plan, la ESPOL puede ser un gran aporte para llevarlo a cabo.

ANEXO A

*Modelo del Sistema de
Información Turístico (SIT)
Eden*

A.1 Modelo de Sistema Información Turístico



ANEXO B

*Manual del Usuario del
Sistema de Información
Turístico (SIT) Eden*

B.1 REQUERIMIENTOS DEL SISTEMA

Requerimientos de Hardware

Las características mínimas de un computador para manejar el sistema son:

- Procesador Pentium
- 32 MB de memoria RAM
- 2 GB de disco duro

Requerimientos de Software

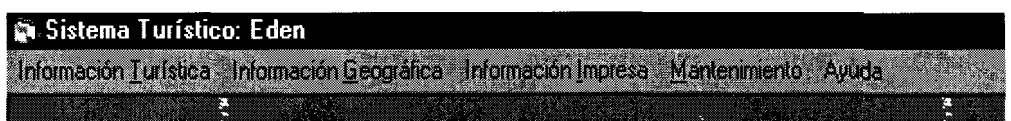
Las características mínimas del software de un computador para manejar el sistema son:

- Windows 95 o superior
- Microsoft Office 97 o superior

B.2 CARACTERISTICAS GENERALES DEL SISTEMA

Menú Principal

El programa tiene un menú principal que siempre esta activo y se encuentra en la parte superior como se muestra a continuación:



Ventanas del Sistema

La mayoría de las ventanas tienen:

- Una tabla de la cual se puede seleccionar un ítem.

INDIC	DESCRIPCIÓN	ESTADO
47	LAVANDERIA	A
52	OTROS RENT A CAR	A
35	PARKINGERO O ESTACIONAMIENTO	A
25	PELLUQUERIA	A
82	PESCA DEPORTIVA	A
▶	PISCINA	A
58	PLAZA DE TOROS	A
41	REFRIGERADORA	A
23	RESTAURANTE	A
91	RIVM SERVICE	A

- Botones para Agregar, Eliminar, Consultar, Cerrar, etc. que siempre estarán en la parte inferior de las ventanas



- Una barra de estado con botones en los extremos. Se le dice barra de estado porque indica la acción que se está ejecutando o el número del registro que está seleccionando. Los botones en los extremos son para recorrer los registros de la lista o tabla.



Sirve para ir al primer registro de la lista



Sirve para ir al registro anterior del registro actual



Sirve para ir al siguiente registro



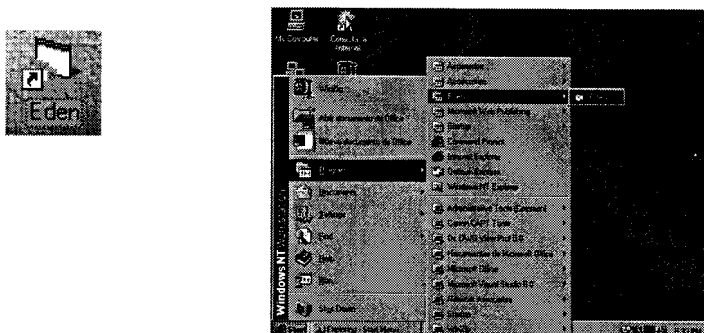
Para ir al ultimo registro de la lista

- Cajas de texto para visualizar o modificar la información

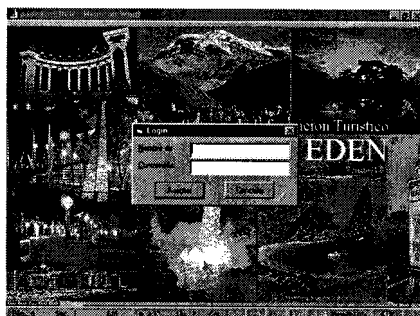
Código:	<input type="text" value="01"/>
Descripción:	<input type="text" value="PISCINA"/>

B.3 INGRESO AL SISTEMA EDEN

Para ingresar al sistema debe hacer doble clic en el icono o escoja de la barra de inicio: Programas / Eden



Se mostrará la siguiente pantalla:

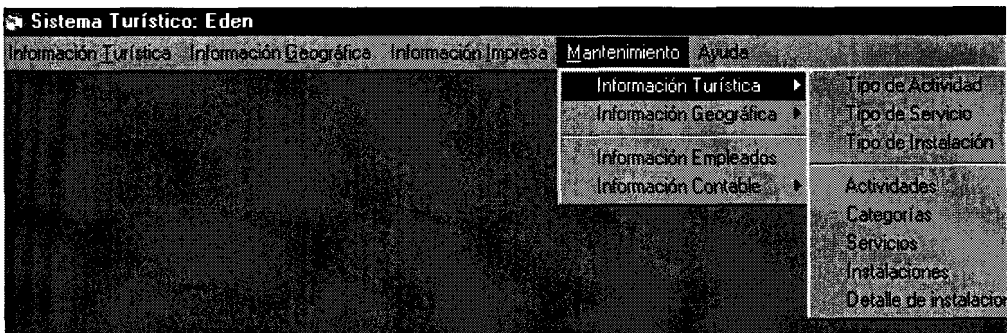


Debe ingresar su usuario y password y luego presionar el botón Aceptar.

B.4 MANTENIMIENTO DE DATOS GENERALES DEL SISTEMA

Información Turística

Dentro del menú *Mantenimiento* de la barra principal, tenemos el menú *Información Turística* del cual se despliega otro menú como indica la figura.



El menú *Información Turística* incluye todos los datos generales que se requieren para llenar la información de las plantas turísticas. Estos datos son:

Tipo de Actividad

Tipo de Servicio

Tipo de Instalación

Actividades

Categorías

Servicios

Instalaciones

Detalle de instalaciones

A continuación se detallan cada uno de estos items.

Tipo de Actividad

Esta parte del sistema permite ingresar los tipos de actividades que pueden tener las ciudades, cantones, provincias, etc.

Tipo de Actividad consiste en una lista en la que cada elemento tiene su respectiva lista de actividades, es decir, por ejemplo, un tipo de actividad es Alojamiento y sabemos que hay diferentes formas de alojamiento como: Hotel, Hostal, Motel, etc. estos últimos son las actividades. También otro tipo de actividad es Comidas y Bebidas dentro del cual existe un conjunto de actividades como: Restaurant, Cafetería, Bar, etc.

A continuación se detalla algunos tipos de actividades con sus respectivas actividades:

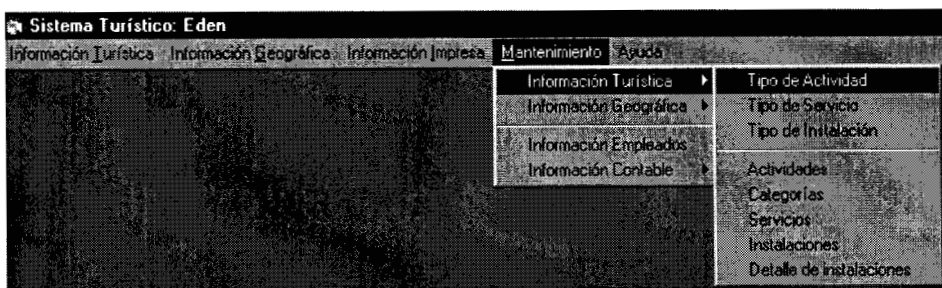
Tipo de Actividad	Actividades
Alojamiento	Albergue Apartamento Turístico Cabaña Hostal Hostal Residencia Hostería Hotel Hotel Apartamento Hotel Residencia

	<p>Motel</p> <p>ciudad Vacacional</p> <p>campamentos turísticos</p> <p>caravandor</p> <p>estación</p> <p>refugio</p>
Comidas y Bebidas	<p>bar</p> <p>restaurant</p> <p>Cafetería</p> <p>Drive Inn</p> <p>Fuente de Soda</p>
Agencias de Viajes	Agencias de Viajes
Diversión y Recreación	<p>Centro de Convención</p> <p>Centro de Recreación Turístico</p> <p>Discoteca</p> <p>Peña</p> <p>Pista de patinaje</p> <p>Sala de baile</p> <p>Balneario</p> <p>Boleras</p> <p>Sala de recepción y banquetes</p>

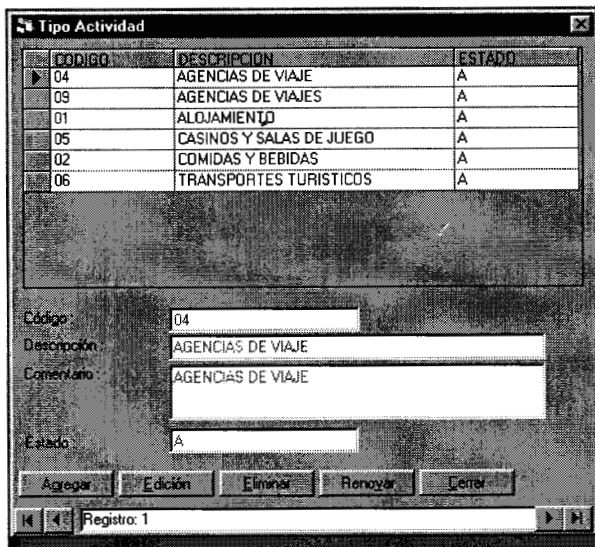
En esta sección sólo veremos los tipos de actividades, más adelante se verá las actividades que tiene cada tipo de actividad.

Para ingresar a esta parte del sistema haga los siguientes pasos:

1. Escoja del menú principal: *Mantenimiento / Información Turística / Tipo de Actividad*.



2. Luego se mostrará la siguiente ventana:

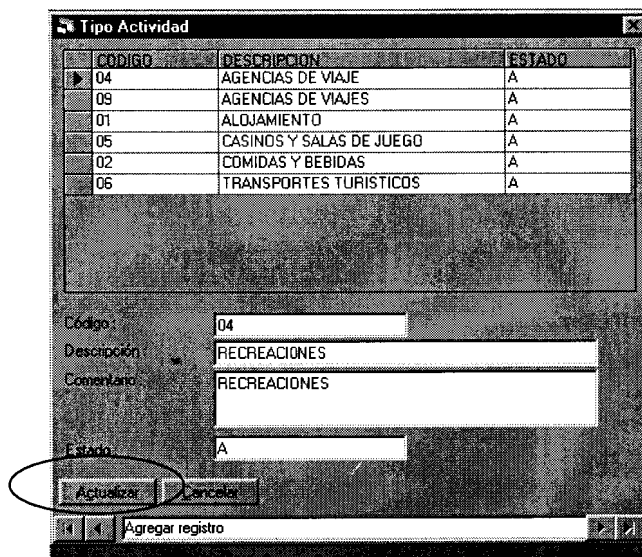


3. Para salir haga clic en el botón que dice: Cerrar.

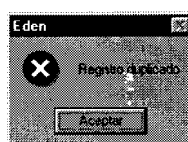
Ingreso Tipo de Actividad

Una vez que está en la ventana Tipo de Actividad que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic con el mouse en el botón que dice: Agregar.
2. Automáticamente, se mostrarán las cajas de texto vacías listas para escribir los datos del nuevo Tipo de Actividad que son: Código, Descripción, Comentario.
3. Una vez que ha ingresado estos datos, haga clic en el botón Actualizar que se indica en el siguiente gráfico.



4. Si el código que ha ingresado ya existe en la lista entonces le saldrá el siguiente mensaje, el cual no le permitirá grabar este nuevo Tipo de Actividad hasta que Ud. cambie el código.

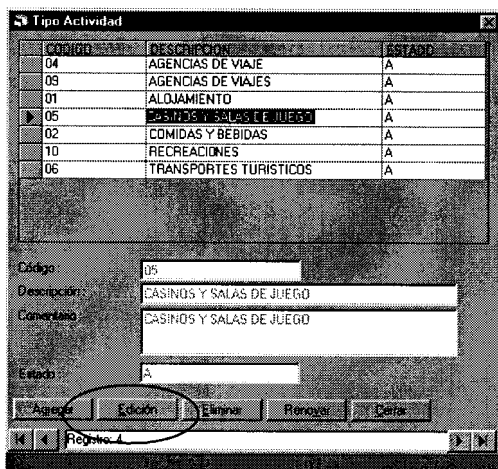


5. Luego de hacer clic en el botón Actualizar (o botón Cancelar si no quiere grabar) automáticamente se verá en la lista el nuevo Tipo de Actividad.

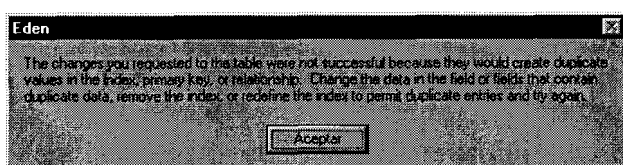
Modificación de Tipo de Actividad

Para modificar los datos de un Tipo de Actividad que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Tipo de Actividad que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de Actividad al cual desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Actividad.
4. Haga clic con el mouse en el botón que dice: Edición como se muestra en el siguiente gráfico:



5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Tipo de Actividad seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Tipo de Actividad

Para sacar del sistema un Tipo de Actividad:

1. Debe ingresar a la ventana Tipo de Actividad que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de Actividad al cual desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Actividad.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Tipo de Actividad eliminado.

Botón Renovar

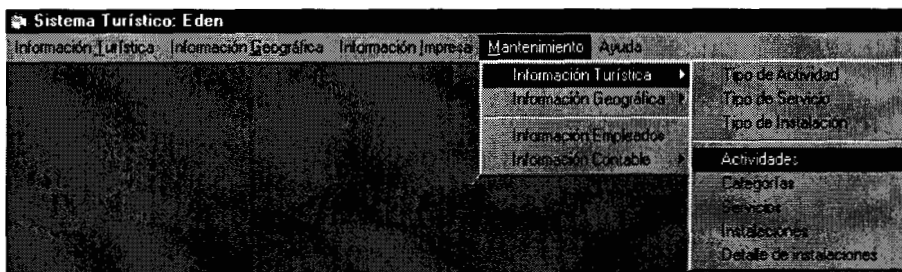
Este botón que está en la ventana Tipo de Actividad sirve, como su nombre lo indica, para renovar o reordenar la lista que se muestra, en orden alfabético.

Actividades

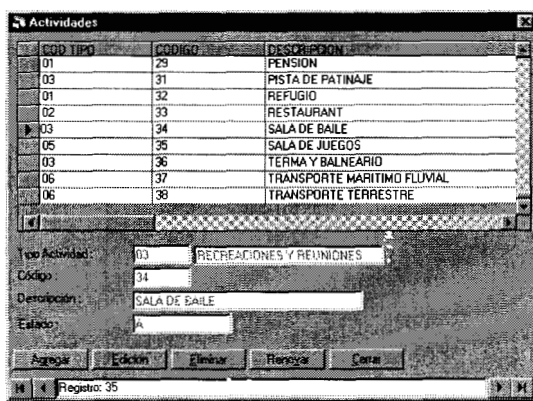
Esto se refiere a las actividades que tiene cada tipo de actividad explicado en la sección anterior. Para el sistema, es necesario tener una lista de actividades de la cual se pueda elegir para registrarlo en la planta turística.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Turística / Actividades*



2. Se mostrará la siguiente ventana:

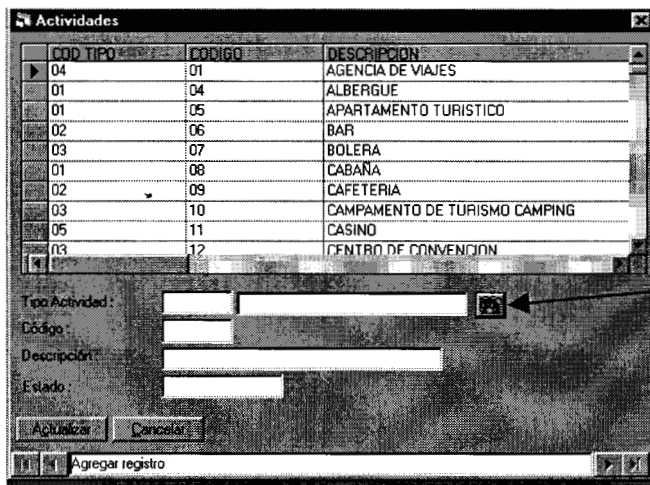


3. Para salir haga clic en el botón que dice: Cerrar

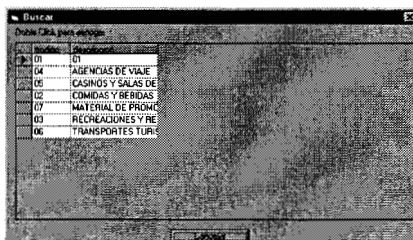
Ingreso de Actividades

Una vez que está en la ventana Actividades que se mostró en el gráfico anterior haga los siguientes pasos:

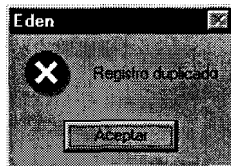
1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar el Tipo de Actividad al que pertenece la actividad que va a ingresar.



3. Luego de presionar el botón se mostrará una lista de tipos de actividades de la cual debe seleccionar uno haciendo doble clic.



4. Luego ingrese el resto de datos que pide: Código, Descripción, Estado
5. Luego de llenar los datos debe hacer clic en el botón Actualizar.
6. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



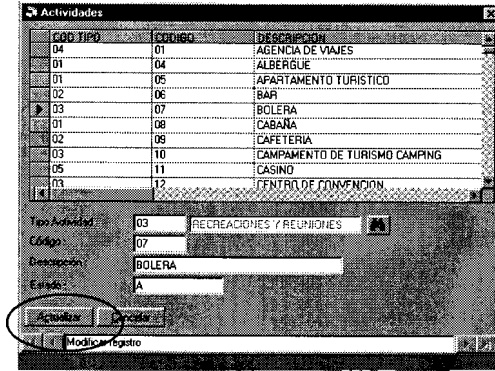
7. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Modificación de Actividades

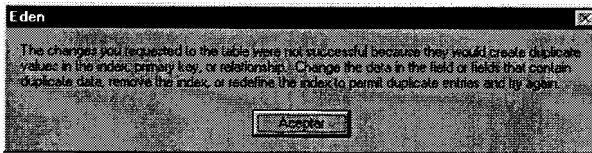
Para modificar los datos de una Actividad que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Actividades que se indica en el inicio de esta sección
2. Escoja de la lista la Actividad que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Actividad.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Actividad seleccionada.

6. Luego de modificar lo que desea, presione el botón Actualizar.



7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Actividades

Para sacar del sistema una Actividad:

1. Debe ingresar a la ventana Actividad que se indica en el inicio de esta sección
2. Escoja de la lista la Actividad que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Actividad.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Actividad eliminada.

Botón Renovar

Este botón que está en la ventana Actividades, sirve como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Categorías

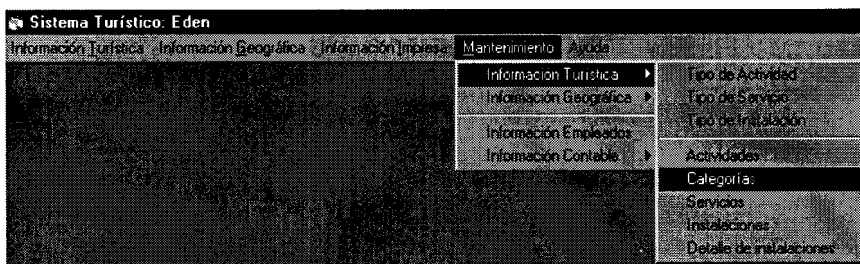
Esto se refiere a un listado de todas las posibles categorías de las plantas turísticas.

A continuación se muestra una lista de algunas categorías que se pueden utilizar para el registro de las plantas turísticas.

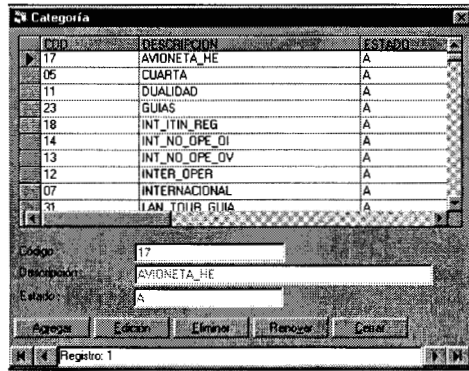
Categorías
Lujo
Primera
Segunda
Tercera
Cuarta
Mayor
Internacional
Operadora
Permanente
Temporal
Dualidad
Películas
Rent a Car, etc.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Turística / Categorías*



2. Se mostrará la siguiente ventana:

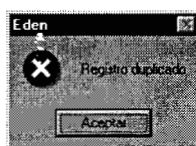


3. Para salir haga clic en el botón que dice: Cerrar

Ingreso Categorías

Una vez que está en la ventana Categoría que se mostró anteriormente, haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Automáticamente podrá escribir en las cajas de texto los datos de la nueva Categoría: Código, Descripción, Estado
3. Luego de llenar los datos debe hacer clic en el botón Actualizar.
4. Si Ud. ha ingresado un código que ya existe, entonces le saldrá el siguiente mensaje que no le permitirá grabar hasta que cambie el código.

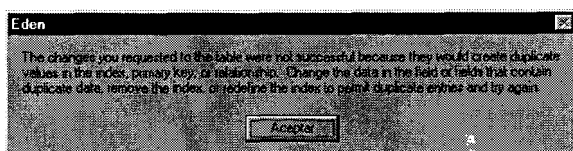


5. Después de haber hecho clic en el botón Actualizar, se mostrará en la lista la categoría ingresada.

Modificación de Categoría

Para modificar una categoría registrada en el sistema haga lo siguiente:

1. Debe ingresar a la ventana Categoría que se indica en el inicio de esta sección
2. Escoja de la lista la Categoría que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Categoría.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Categoría seleccionada.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.





8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Categoría



Para borrar del sistema una Categoría:

1. Debe ingresar a la ventana Categoría que se indica en el inicio de esta sección
2. Escoja de la lista la Categoría que va a eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Servicio.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Categoría eliminada.

Botón Renovar

Este botón que está en la ventana Categoría sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.





Tipo de Servicio

Esto se refiere a un listado de todos los tipos de servicios que puede ofrecer una planta turística.

A continuación se detalla algunos tipos de servicios:

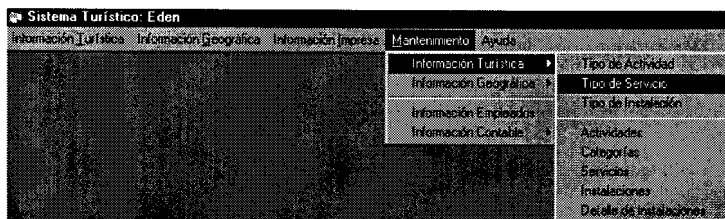
Tipos de Servicios
Aire Acondicionado y/o Calefacción
Bar
Botes
Boutique
Cafetería
Caja de seguridad
Canchas Deportivas
Casinos
Discoteca
Enfermería
Galería
Gimnasio
Habitación TV Cable
Hidromasajes
Jacuzzi
Juegos, etc. etc.



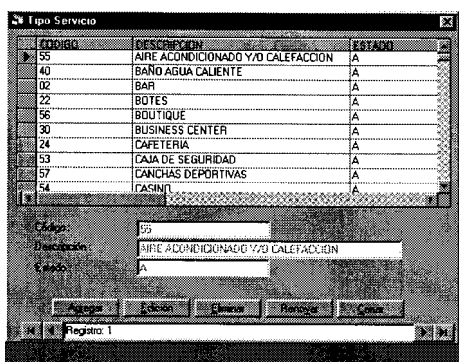
Para ingresar a esta área del sistema debe:



1. Escoger del menú principal: *Mantenimiento / Información Turística / Tipo de Servicio*



2. Se mostrará la siguiente ventana:

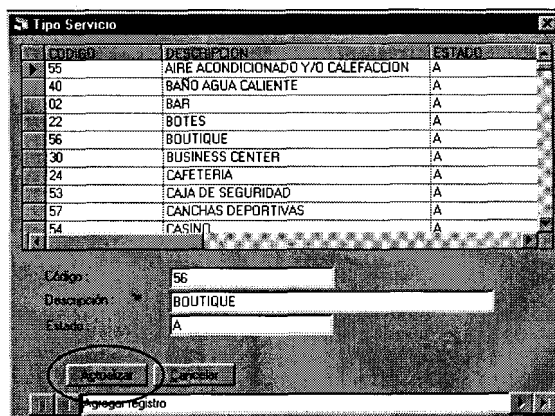


3. Para salir haga clic en el botón que dice: Cerrar

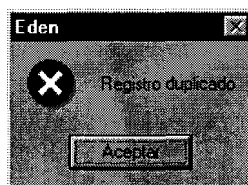
Ingreso Tipo de Servicio

Una vez que está en la ventana Tipo de Servicio que se mostró en el Último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Automáticamente podrá escribir en las cajas de texto los datos del nuevo Tipo de Servicio: Código, Descripción, Estado
3. Luego de llenar los datos debe hacer clic en el botón Actualizar, como se muestra en el siguiente gráfico.



4. Si Ud. ha ingresado un código que ya existe, entonces le saldrá el siguiente mensaje que no le permitirá grabar hasta que cambie el código.

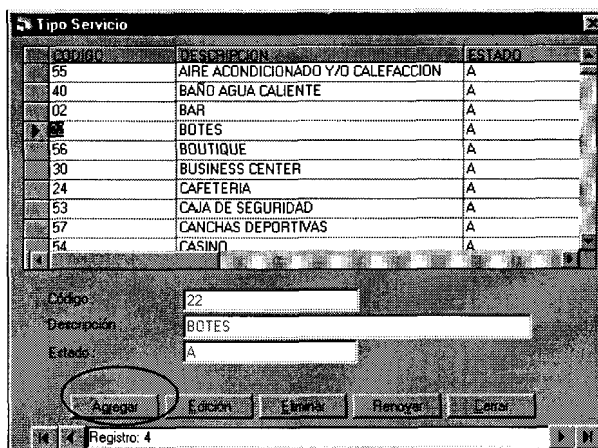


- Después de haber hecho clic en el botón Actualizar, se mostrará en la lista el nuevo Tipo de Servicio.

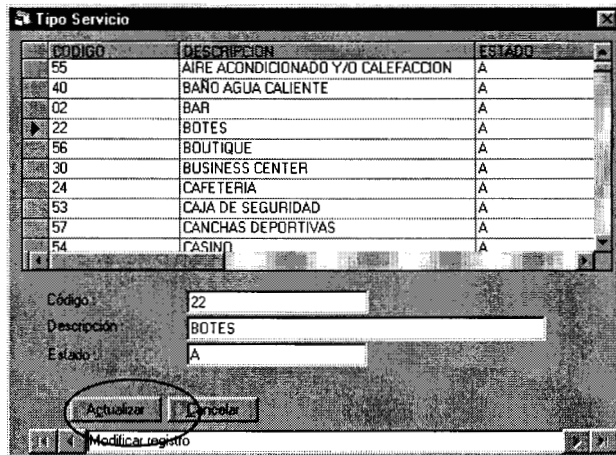
Modificación de Tipo de Servicio

Para modificar los datos de un Tipo de Servicio que tiene el sistema haga lo siguiente:

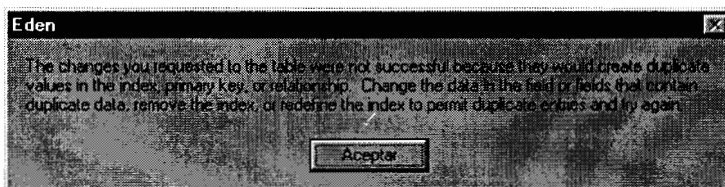
- Debe ingresar a la ventana Tipo de Servicio que se indica en el inicio de esta sección
- Escoja de la lista el Tipo de Servicio al cual desea modificar, haciendo un clic con el mouse.
- Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Servicio.
- Haga clic con el mouse en el botón que dice: Edición como se muestra en el siguiente gráfico:



- Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Tipo de Servicio seleccionado.
- Luego de modificar lo que desea, presione el botón Actualizar.



- Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



1. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Tipo de Servicio

Para sacar del sistema un Tipo de Servicio:

1. Debe ingresar a la ventana Tipo de Servicio que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de Servicio al cual desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Servicio.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Tipo de Servicio eliminado.

Botón Renovar

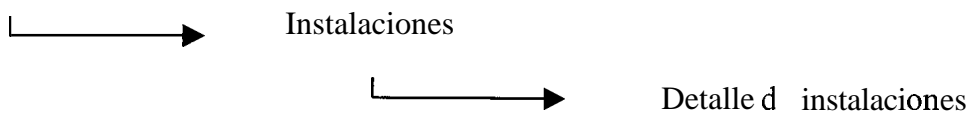
Este botón que está en la ventana Tipo de Servicio sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Tipo de Instalación

Esta parte del sistema se refiere a todos los tipos de instalación que puede tener una planta turística. En el sistema, es necesario tener una lista de tipos de instalación de la cual se pueda elegir para registrarlo en la planta turística.

Con respecto a las instalaciones el sistema las organiza de la siguiente forma:

Tipo de Instalación



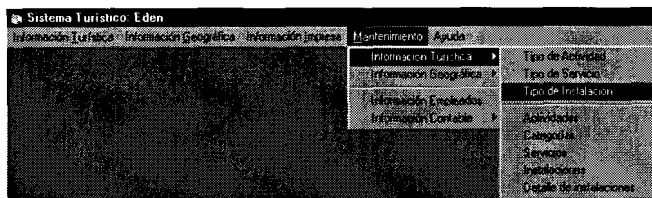
A continuación se muestra un ejemplo de algunos tipos de instalaciones:

Tipo Instalación	Instalaciones	Detalle de Instalaciones
Recreacionales	Discoteca	CBI
Comedores y Bares	Restaurante Cafetería Fuente de Soda Bar	SBP BC Teléfono Fax Aire Acondicionado
Habitaciones	Sencilla Matrimonial Doble Triple Cuadruple Suite	Ascensores Insonorización Frigobar Televisor Hidromasaje Sauna

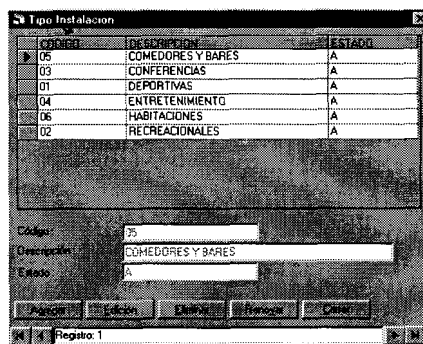
La lista de Detalle de Instalaciones es la misma para cada instalación. Primero se explicará el manejo de tipos de instalaciones, luego las instalaciones y después, el detalle de instalaciones.

Para ingresar al área del Tipo de Instalaciones debe:

1. Escoger del menú principal: *Mantenimiento / Información Turística / Tipo de Instalación*



2. Se mostrará la siguiente ventana:

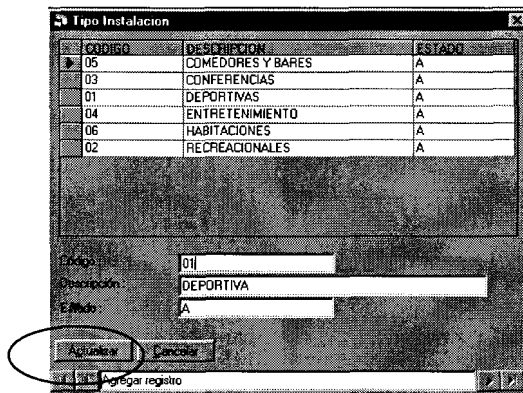


3. Para salir haga clic en el botón que dice: Cerrar

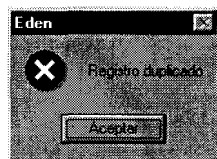
Ingreso Tipo de Instalación

Una vez que está en la ventana Tipo de Instalación que se mostró en el Último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Automáticamente podrá escribir en las cajas de texto los datos del nuevo Tipo de Instalación: Código, Descripción, Estado
3. Luego de llenar los datos debe hacer clic en el botón Actualizar, como se muestra en el siguiente gráfico.



4. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.

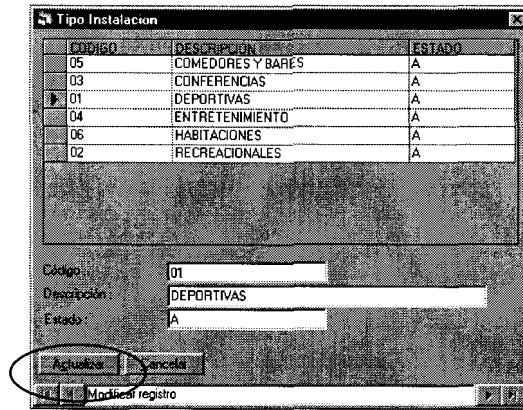


5. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

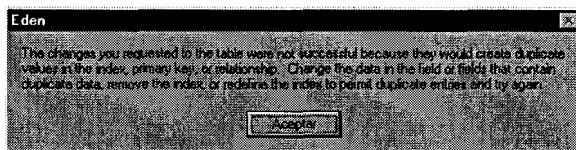
Modificación de Tipo de Instalación

Para modificar los datos de un Tipo de Instalación que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Tipo de Instalación que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de Instalación al cual desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Instalación.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Tipo de Instalación seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.



7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación Tipo de Instalación

Para sacar del sistema un Tipo de Instalación:

1. Debe ingresar a la ventana Tipo de Instalación que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de Instalación al cual desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de Instalación.

4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Tipo de Instalación eliminado.

Botón Renovar

Este botón que está en la ventana Tipo de Instalación sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

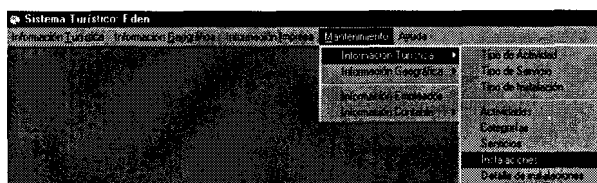
Instalación

Esto se refiere a un listado de todas las instalaciones que puede ofrecer una planta turística.

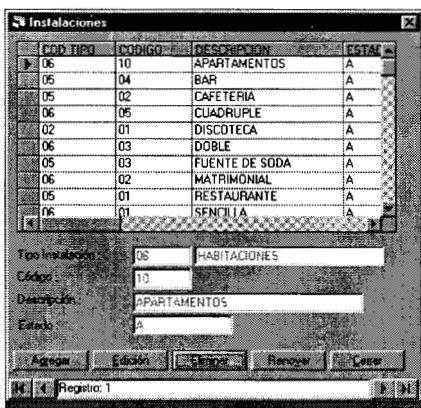
En la sección anterior se explica y se da ejemplo de Instalaciones.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Turística / Instalaciones*



2. Se mostrará la siguiente ventana:

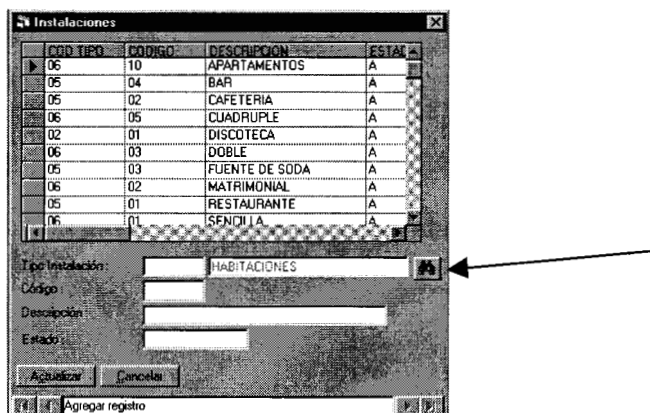


3. Para salir haga clic en el botón que dice: Cerrar

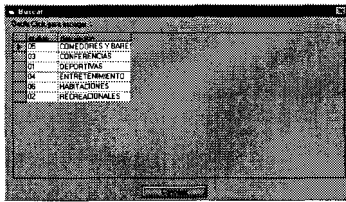
Ingreso Instalaciones

Una vez que está en la ventana Instalaciones que se mostró en el último gráfico haga los siguientes pasos:

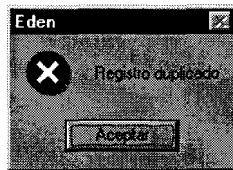
1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar el Tipo de Instalación al que pertenece la instalación que va a ingresar.



3. Luego de presionar el botón se mostrará una lista de tipos de instalaciones de la cual debe seleccionar uno haciendo doble clic.



4. Luego ingrese el resto de datos que pide: Código, Descripción, Estado
5. Luego de llenar los datos debe hacer clic en el botón Actualizar.
6. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



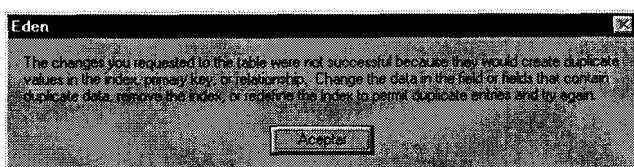
7. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Modificación de Instalación

Para modificar los datos de una Instalación que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Instalaciones que se indica en el inicio de esta sección
2. Escoja de la lista la Instalación que desea modificar, haciendo un clic con el mouse.

3. Automáticamente se mostrará en las cajas de texto la información de esa Instalación.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Instalación seleccionada.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Instalaciones

Para sacar del sistema una Instalación:

1. Debe ingresar a la ventana Instalaciones que se indica en el inicio de esta sección
2. Escoja de la lista la Instalación que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Instalación.

4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Instalación eliminada.

Botón Renovar

Este botón que está en la ventana Instalaciones sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

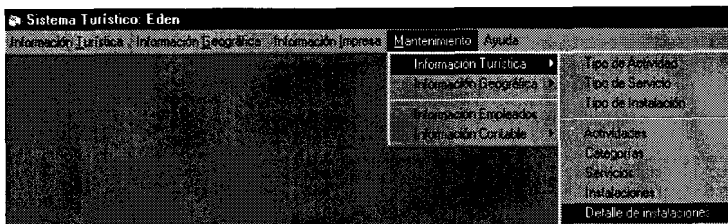
Detalle de Instalación

Esto se refiere a un listado de todas las instalaciones que puede ofrecer una planta turística.

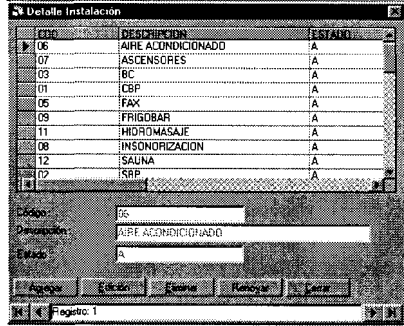
En la sección anterior se explica y se da ejemplo de Instalaciones.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Turística / Instalaciones*



2. Se mostrará la siguiente ventana:

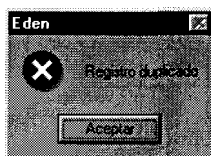


3. Para salir haga clic en el botón que dice: Cerrar

Ingreso Detalle de Instalaciones

Una vez que está en la ventana Detalle Instalaciones que se mostró en el Último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Automáticamente podrá escribir en las cajas de texto los datos del nuevo Detalle de Instalación: Código, Descripción, Estado
3. Luego de llenar los datos debe hacer clic en el botón Actualizar.
4. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.

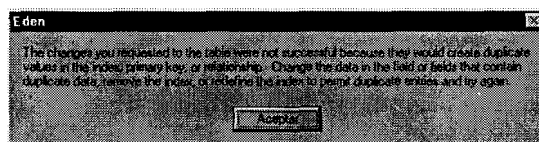


5. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Modificación Detalle de Instalaciones

Para modificar un Detalle de Instalación que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Detalle Instalaciones que se indica en el inicio de esta sección
2. Escoja de la lista el Detalle Instalación que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Detalle Instalación.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Instalación seleccionada.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Detalle Instalaciones

Para sacar del sistema un Detalle Instalación:

1. Debe ingresar a la ventana Detalle Instalaciones que se indica en el inicio de esta sección
2. Escoja de la lista el que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Detalle Instalación.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista.

Botón Renovar

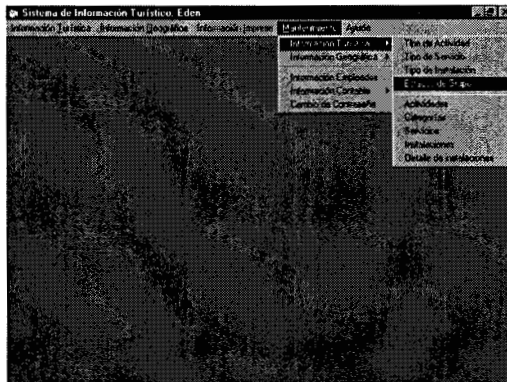
Este botón que está en la ventana Instalaciones sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Estados de Grupo

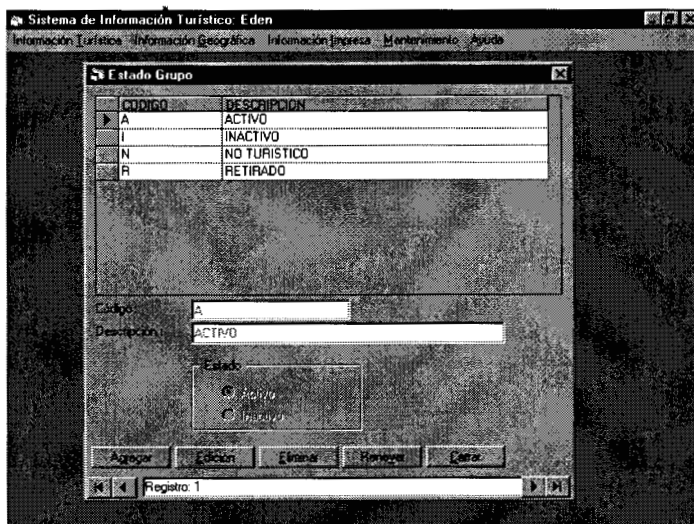
Esta parte del sistema se refiere a un listado de todos los posibles estados de una planta turística. Estos estados pueden ser: Activo, Inactivo, Retirado, No turístico, etc.

Para ingresar a esta parte del sistema haga los siguientes pasos:

1. Escoja del menú principal: *Mantenimiento / Información Turística/Estados de Grupo.*



2. Luego se mostrará la siguiente ventana:

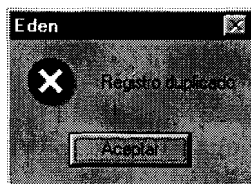


3. Para salir haga clic en el botón que dice: Cerrar.

Ingreso Estados de Grupo

Una vez que está en la ventana Estados de Grupo que se mostró en el Último gráfico haga los siguientes pasos:

1. Haga clic con el mouse en el botón que dice: Agregar.
2. Automáticamente, se mostrarán las cajas de texto vacías listas para escribir los datos del nuevo Estado de Grupo que son: Código, Descripción.
3. Una vez que ha ingresado estos datos, haga clic en el botón Actualizar
4. Si el código que ha ingresado ya existe en la lista entonces le saldrá el siguiente mensaje, el cual no le permitirá grabar este nuevo Tipo de Actividad hasta que Ud. cambie el código.

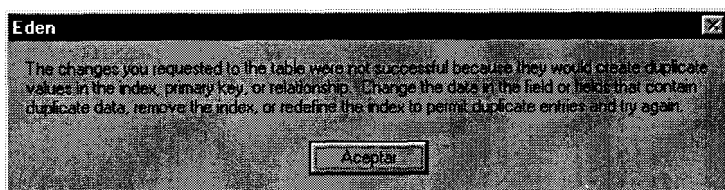


5. Luego de hacer clic en el botón Actualizar (o botón Cancelar si no quiere grabar) automáticamente se verá en la lista el nuevo Estado de Grupo.

Modificación de Estado de Grupo

Para modificar los datos de un Estado de Grupo que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Estado de Grupo que se indica en el inicio de esta sección
2. Escoja de la lista el Estado de Grupo al cual desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Estado de Grupo.
4. Haga clic con el mouse en el botón que dice: Edición.
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Estado de Grupo seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. **Si** le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Estado de Grupo

Para sacar del sistema un Estado de Grupo:

1. Debe ingresar a la ventana Estado de Grupo que se indica en el inicio de esta sección
2. Escoja de la lista el Estado de Grupo al cual desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Estado de Grupo.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Estado de Grupo eliminado.

Botón Renovar

Este botón que está en la ventana Estado de Grupo sirve, como su nombre lo indica, para renovar o reordenar la lista que se muestra, en orden alfabético.

Información Geográfica

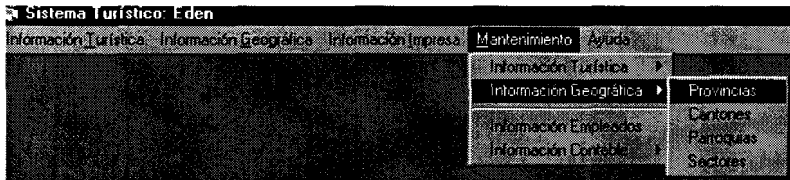
Esto se refiere a todas las provincias, cantones, parroquias que maneja el sistema Eden.

Provincias

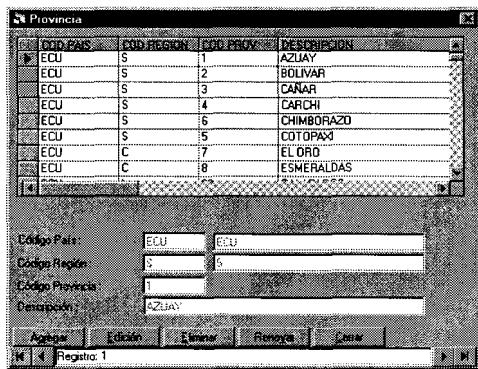
Esta parte del sistema se refiere a un listado de todas las provincias que se manejan en el sistema Eden.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Geográfica/ Provincias*



2. Se mostrará la siguiente ventana:

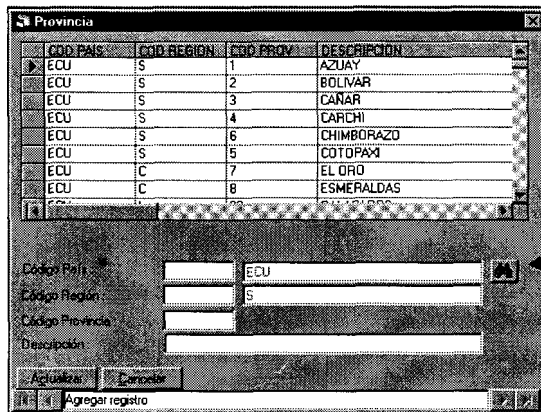


3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Provincias

Una vez que está en la ventana Provincias que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar el País.



3. Luego de presionar el botón se mostrará una lista de países de la cual debe seleccionar uno haciendo doble clic.
4. Luego, de la misma manera es para código de región
5. Para poner código de provincia asegúrese de no repetirlo.
6. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código de la provincia, porque está repetido.

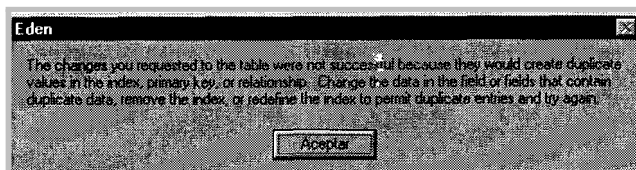


7. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha ingresado.

Modificación de Provincias

Para modificar los datos de una Provincia que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Provincia que se indica en el inicio de esta sección
2. Escoja de la lista la Provincia que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de la Provincia.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Provincia seleccionada.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Provincias

Para sacar del sistema una Provincia:

1. Debe ingresar a la ventana Provincias que se indica en el inicio de esta sección
2. Escoja de la lista la Provincia que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Provincia.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Provincia eliminada.

Botón Renovar

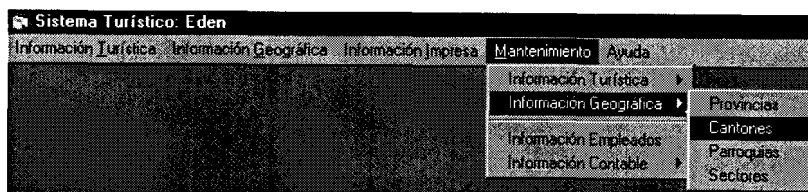
Este botón que está en la ventana Provincias sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Cantones

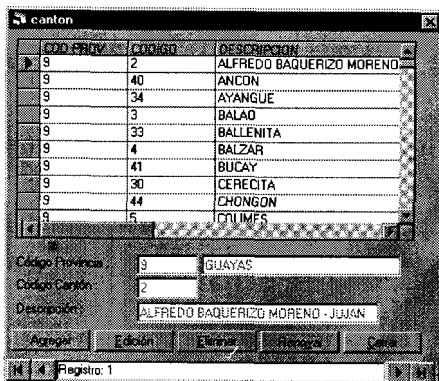
Esta parte del sistema se refiere a un listado de todos los cantones que se manejan en el sistema Eden.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Geográfica/ Cantones*



2. Se mostrará la siguiente ventana:

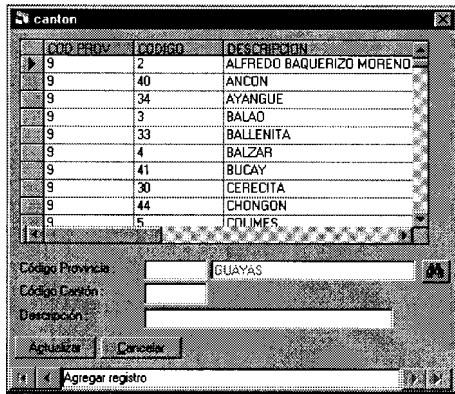


3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Cantones

Una vez que está en la ventana Canton que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar el País.



3. Luego de presionar el botón se mostrará una lista de cantones de la cual debe seleccionar uno haciendo doble clic.
4. Para poner código de cantón asegúrese de no repetirlo.
5. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código del cantón, porque está repetido.

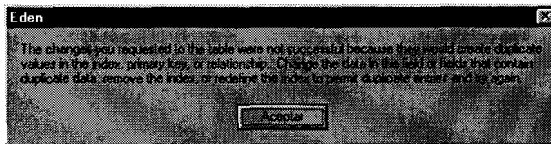


6. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha ingresado.

Modificación de Cantones

Para modificar los datos de un Cantón que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Cantón que se indica en el inicio de esta sección
2. Escoja de la lista el Cantón que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información del Cantón.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Cantón seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Cantones

Para sacar del sistema un Cantón:

1. Debe ingresar a la ventana Cantón que se ~~indica~~ ^{indica} en el inicio de esta sección
2. Escoja de la lista el Cantón que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Cantón.

4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Cantón eliminado.

Botón Renovar

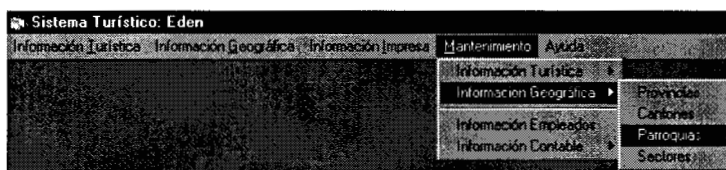
Este botón que está en la ventana Cantón sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Parroquias

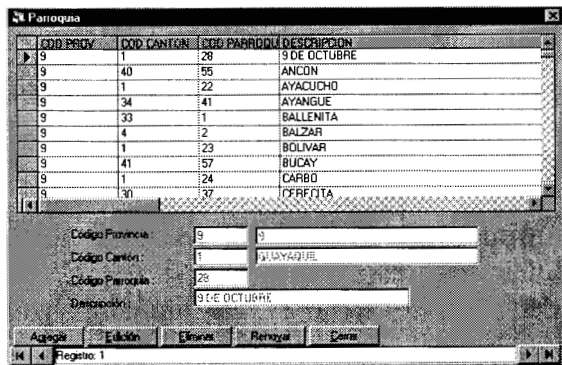
Esta parte del sistema se refiere a un listado de todas las parroquias que se manejan en el sistema Eden.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Geográfica/ Parroquias*



2. Se mostrará la siguiente ventana:

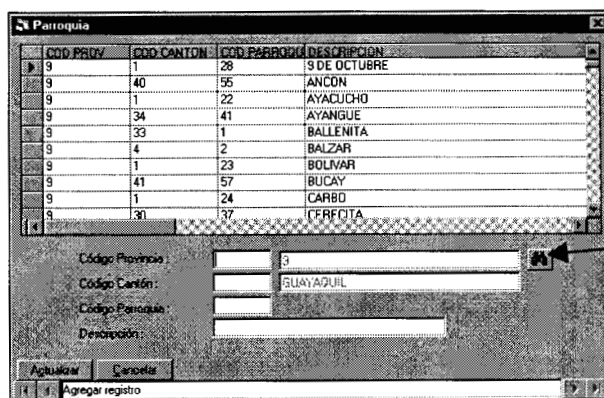


3. Para salir haga clic en el botón que dice: Cerrar

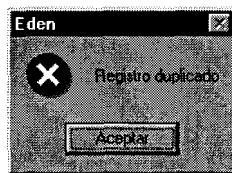
Ingreso de Parroquias

Una vez que está en la ventana Parroquias que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar la Provincia.



3. Luego de presionar el botón se mostrará una lista de provincias de la cual debe seleccionar una haciendo doble clic.
4. Luego, de la misma manera es para código de cantón
5. Para poner código de parroquia asegúrese de no repetirlo.
6. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código de la provincia, porque está repetido.



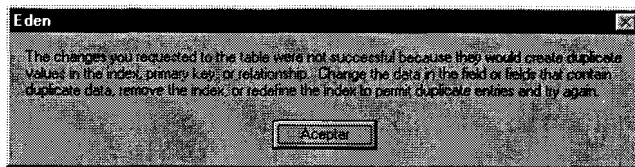
7. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha ingresado.

Modificación de Parroquias

Para modificar los datos de una Parroquia que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Parroquia que se indica en el inicio de esta sección
2. Escoja de la lista la Parroquia que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de la Parroquia.
4. Haga clic con el mouse en el botón que dice: Edición

5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Parroquia seleccionada.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Parroquias

Para sacar del sistema una Parroquia:

1. Debe ingresar a la ventana Parroquia que se indica en el inicio de esta sección
2. Escoja de la lista la Parroquia que desea eliminar, haciendo un clic con el mouse.

3. Automáticamente se mostrará en las cajas de texto la información de esa Parroquia.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Parroquia eliminada.

Botón Renovar

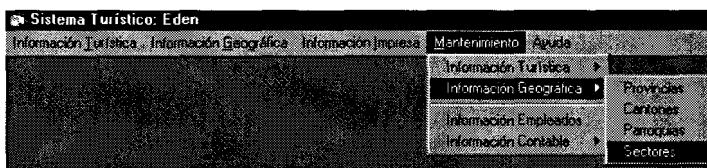
Este botón que está en la ventana Parroquia sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Sectores

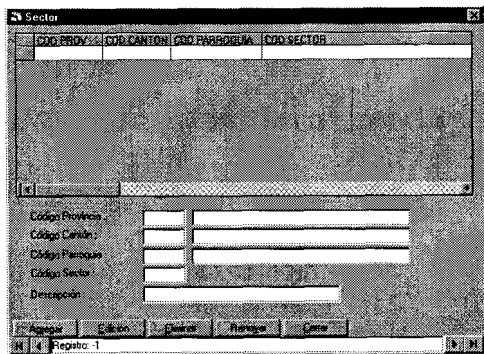
Esta parte del sistema se refiere a un listado de todos los sectores de las parroquias que se manejan en el sistema Eden.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Geográfica/ Sectores*



2. Se mostrará la siguiente ventana:

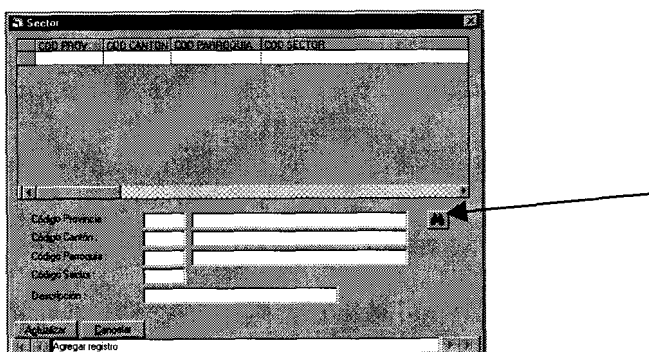


3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Sectores

Una vez que está en la ventana Sectores que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Se mostrará la siguiente ventana, en la cual debe presionar el botón señalado para seleccionar la Provincia.



3. Luego de presionar el botón se mostrará una lista de provincias de la cual debe seleccionar una haciendo doble clic.
4. Luego, de la misma manera es para cantón y parroquia.
5. Para poner código de sector asegúrese de no repetirlo.
6. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código del sector, porque está repetido.



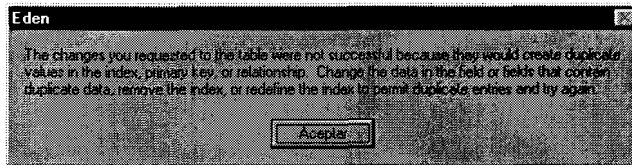
7. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha ingresado.

Modificación de Sectores

Para modificar los datos de un Sector que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Sector que se indica en el inicio de esta sección
2. Escoja de la lista el Sector que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información del Sector.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Sector seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.

7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Sectores

Para sacar del sistema un Sector:

1. Debe ingresar a la ventana Sector que se indica en el inicio de esta sección
2. Escoja de la lista el Sector que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Sector.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Sector eliminada.

Botón Renovar

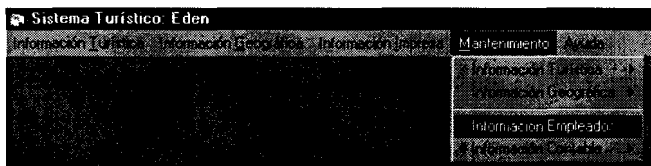
Este botón que está en la ventana Sector sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Información Empleados

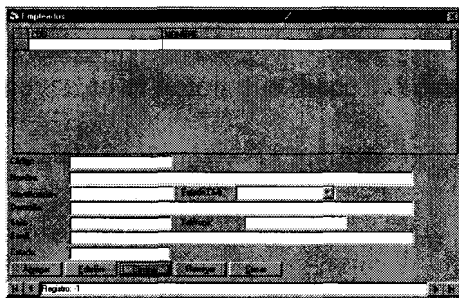
El sistema también permite guardar la información de los empleados de la Institución, en este caso del Ministerio de Turismo.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Empleados*



2. Se mostrará la siguiente ventana:



3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Empleados

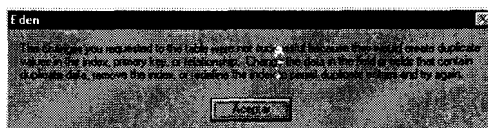
Una vez que está en la ventana Empleados que se mostró en el Último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Las cajas de texto estarán listas para ingresar todos los datos requeridos.
3. Luego presione el botón Actualizar para grabar la información o caso contrario presione el botón Cancelar.

Modificación de Empleados

Para modificar los datos de un Empleado que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Empleados que se indica en el inicio de esta sección
2. Escoja de la lista el Empleado que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información del Empleado.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Empleado seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Empleados

Para sacar del sistema un Empleado:

1. Debe ingresar a la ventana Empleados que se indica en el inicio de esta sección
2. Escoja de la lista el Empleado que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Empleado.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Empleado eliminado.

Botón Renovar

Este botón que está en la ventana Empleados sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Información Contable

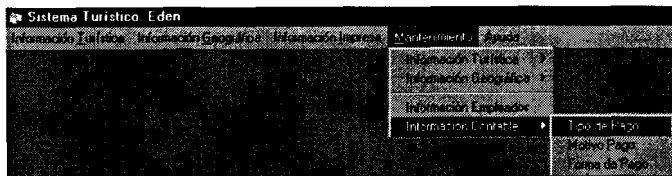
Para tener un control de los pagos que realizan las plantas turísticas el sistema tiene esta área que consta de un listado de tipo de pago, forma de pago y motivo de pago.

Tipo de pago

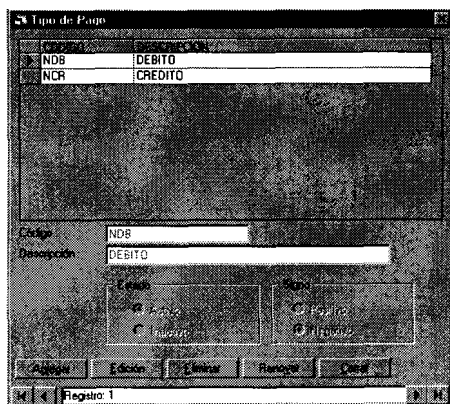
El sistema requiere un listado de los posibles tipos de pago que hace una planta turística.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Contable / Tipo de pago*



2. Se mostrará la siguiente ventana:



3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Tipo de pago

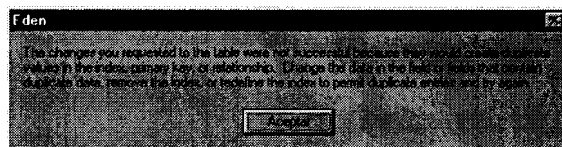
Una vez que está en la ventana Tipo de pago que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: *Agregar*
2. Las cajas de texto estarán listas para ingresar todos los datos requeridos.
3. Luego presione el botón *Actualizar* para grabar la información o caso contrario presione el botón *Cancelar*.

Modificación Tipo de pago

Para modificar un Tipo de pago que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Tipo de pago que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de pago que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información del Tipo de pago.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos del Tipo de pago seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Tipo de pago

Para sacar del sistema un Tipo de pago:

1. Debe ingresar a la ventana Tipo de pago que se indica en el inicio de esta sección
2. Escoja de la lista el Tipo de pago que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Tipo de pago.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Tipo de pago eliminado.

Botón Renovar

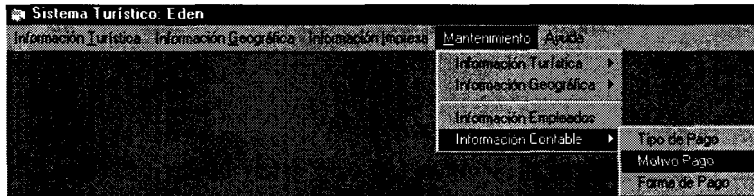
Este botón que está en la ventana Tipo de pago sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Motivo pago

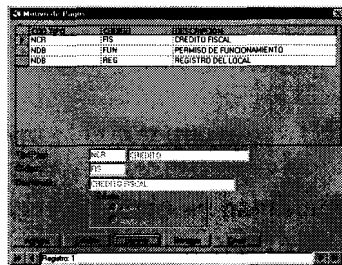
El sistema requiere un listado de los posibles motivos de pago que hace una planta turística.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Contable / Motivo pago*



2. Se mostrará la siguiente ventana:



3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Motivo pago

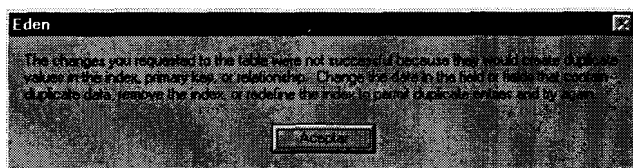
Una vez que está en la ventana Motivo de pago que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Las cajas de texto estarán listas para ingresar todos los datos requeridos.
3. Luego presione el botón Actualizar para grabar la información o caso contrario presione el botón Cancelar.

Modificación de Motivo pago

Para modificar un Motivo de pago que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Motivo de pagos que se indica en el inicio de esta sección
2. Escoja de la lista el Motivo de pago que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información del Motivo de pago.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto *se* habilitarán para que pueda modificar los datos del Motivo de pago seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque-está repetido.



8. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Motivo pago

Para sacar del sistema un Motivo de pago:

1. Debe ingresar a la ventana Motivo de pago que se indica en el inicio de esta sección
2. Escoja de la lista el Motivo de pago que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de ese Motivo de pago.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista el Motivo de pago eliminado.

Botón Renovar

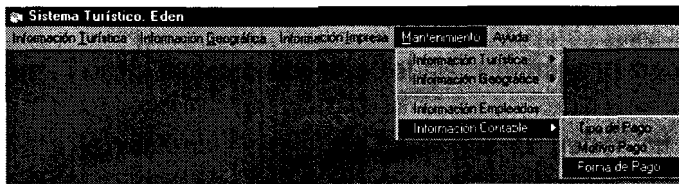
Este botón que está en la ventana Motivo de pago sirve, como su nombre lo indica, para renovar o reordenar la lista, en orden alfabético.

Forma de pago

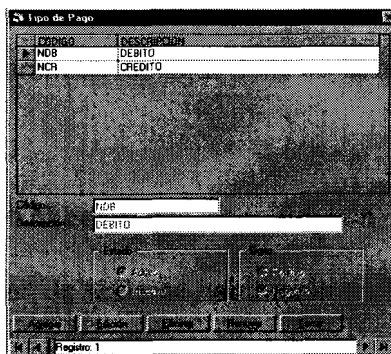
El sistema requiere un listado de todas las posibles formas de pago que hace una planta turística.

Para ingresar a esta área del sistema debe:

1. Escoger del menú principal: *Mantenimiento / Información Contable / Forma de pago*



2. *Se* mostrará la siguiente ventana:



3. Para salir haga clic en el botón que dice: Cerrar

Ingreso de Forma de pago

Una vez que está en la ventana Forma de pago que se mostró en el último gráfico haga los siguientes pasos:

1. Haga clic en el botón: Agregar
2. Las cajas de texto estarán listas para ingresar todos los datos requeridos.
3. Luego presione el botón Actualizar para grabar la información o caso contrario presione el botón Cancelar.

Modificación de Forma de pago

Para modificar una Forma de pago que tiene el sistema haga lo siguiente:

1. Debe ingresar a la ventana Forma de pago que se indica en el inicio de esta sección
2. Escoja de la lista la Forma de pago que desea modificar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de la Forma de pago.
4. Haga clic con el mouse en el botón que dice: Edición
5. Automáticamente las cajas de texto se habilitarán para que pueda modificar los datos de la Forma de pago seleccionado.
6. Luego de modificar lo que desea, presione el botón Actualizar.
7. Si le sale el siguiente mensaje, esto le indica que no le permitirá grabar hasta que cambie el código, porque está repetido.



1. Luego de hacer clic en el botón Actualizar, podrá ver en la lista y en las cajas de texto lo que ha modificado.

Eliminación de Forma de pago

Para sacar del sistema una Forma de pago:

1. Debe ingresar a la ventana Forma de pago que se indica en el inicio de esta sección
2. Escoja de la lista la Forma de pago que desea eliminar, haciendo un clic con el mouse.
3. Automáticamente se mostrará en las cajas de texto la información de esa Forma de pago.
4. Haga clic con el mouse en el botón que dice: Eliminar y en ese momento se quitará de la lista la Forma de pago eliminada.

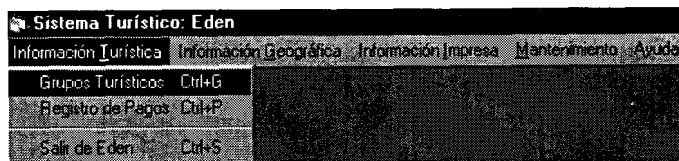
B.5 MANTENIMIENTO DE PLANTA TURISTICA

Se denomina Planta Turística a todo establecimiento que debe tener el permiso de funcionamiento otorgado por el Ministerio de Turismo.

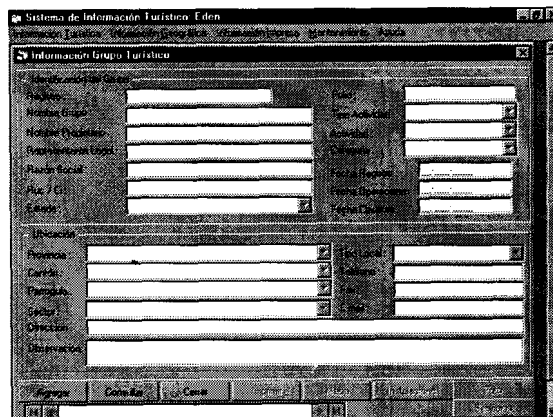
Ingreso de la Planta Turística

Para ingresar al sistema todos los datos de un establecimiento, requeridos por el Ministerio de Turismo, haga los siguientes pasos:

1. Escoger del menú principal: *Información Turística / Grupos Turísticos*



2. Se mostrará la siguiente ventana:



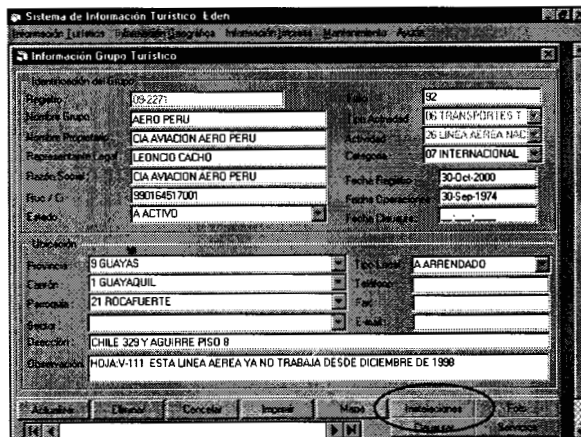
3. Haga clic en el botón: Agregar
4. Automáticamente podrá escribir en las cajas de texto los datos de la nueva Planta Turística
5. Debe llenar todos los campos. Los que pueden quedar en blanco son: E-mail, fax, teléfono, sector y observaciones. En los campos: Fecha Registro, Fecha

Operaciones, Fecha Clausura el formato es: dd-mm-aaaa. Por ejemplo: 22-Dec-2000. Se debe escribir el mes en ingles.

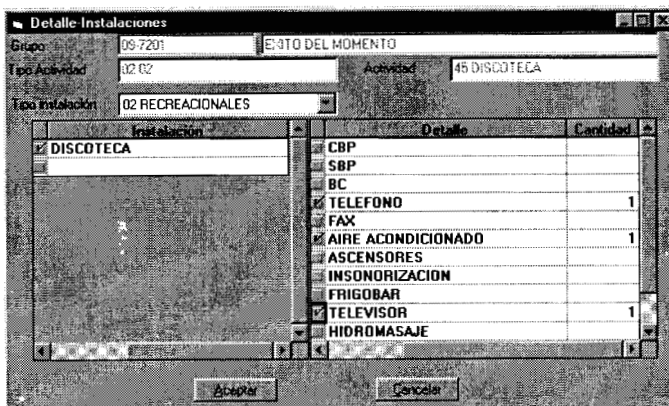
6. Luego de llenar todos los campos, presione el botón: Actualizar, para que se guarde los datos de la planta turística.

Para registrar las instalaciones de una planta turística debe:

1. Hacer la búsqueda de la planta turística. (Se explicará más adelante cómo realizar la búsqueda).
2. Ya con los datos de la planta turística mostrados en pantalla presione el botón: Instalaciones que se señala en la siguiente ventana:



3. Luego se mostrará otra ventana en la cual tiene que elegir el tipo de instalación, las instalaciones y los detalles de instalación (en ese orden) como se muestra:



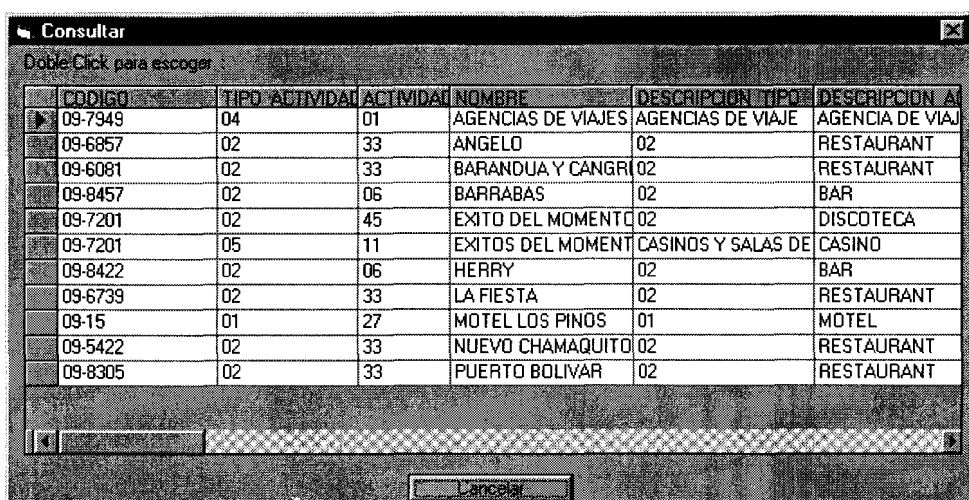
4. Se escoge las instalaciones y los detalles de instalación haciendo clic en el recuadro que está a la izquierda de cada ítem de las listas. Se muestran vistos color rojo indicando que han sido seleccionados.
5. Haga clic en el botón Aceptar para grabar o en el botón Cancelar para salir sin grabar.

Para registrar los servicios de una planta turística es de igual forma que las instalaciones a diferencia que se presiona el botón Servicios en lugar del botón Instalaciones.

Búsqueda de una Planta Turística

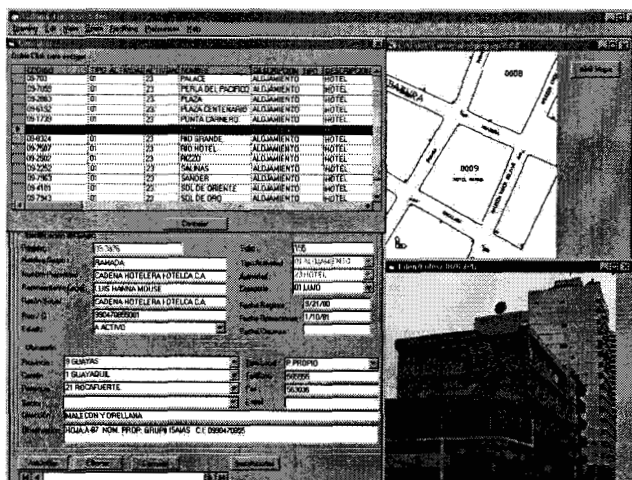
Para encontrar una Planta Turística, haga los siguientes pasos:

1. Haga los dos primeros pasos de la sección ingreso de planta turística
2. Haga clic con el mouse en el botón: Consultar
3. En ese momento, Ud. puede escribir cualquiera de los datos de la planta turística que busca, para hacer una búsqueda rápida, y luego presionar el botón: Ejecutar. En caso de que no recuerde ningún dato, simplemente presione el botón Ejecutar y se mostrará el listado de todas las plantas turísticas ingresadas al sistema en orden alfabético por nombre del establecimiento, como se muestra a continuación.



CODIGO	TIPO ACTIVIDAD	ACTIVIDAD	NOMBRE	DESCRIPCIÓN TIPO	DESCRIPCIÓN ACTIVIDAD
09-7949	04	01	AGENCIAS DE VIAJES	AGENCIAS DE VIAJE	AGENCIA DE VIAJE
09-6857	02	33	ANGELO	02	RESTAURANT
09-6081	02	33	BARANDUA Y CANGRI	02	RESTAURANT
09-8457	02	06	BARRABAS	02	BAR
09-7201	02	45	EXITO DEL MOMENTO	02	DISCOTECA
09-7201	05	11	EXITOS DEL MOMENTO	CASINOS Y SALAS DE	CASINO
09-8422	02	06	HERRY	02	BAR
09-6739	02	33	LA FIESTA	02	RESTAURANT
09-15	01	27	MOTEL LOS PINOS	01	MOTEL
09-5422	02	33	NUEVO CHAMAQUITO	02	RESTAURANT
09-8305	02	33	PUERTO BOLIVAR	02	RESTAURANT

4. Luego de localizar el establecimiento, hacer doble clic en el código para ver todos los datos en pantalla.
5. También puede ver el mapa y la foto haciendo clic en el botón respectivo de la forma.

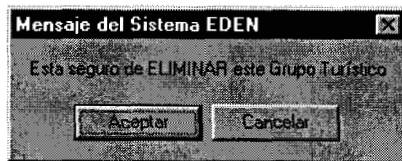


Modificación de Datos de una Planta Turística

1. Localice la Planta turística
2. Una vez mostrados los datos en pantalla, puede modificar todos los campos posibles.
3. Haga clic en el botón Actualizar para grabar los cambios o Cancelar para salir sin grabar.
4. Haga clic en el botón Cerrar para salir de esta ventana.

Eliminación de una Planta Turística

1. Localice la Planta turística
2. Una vez mostrados los datos en pantalla, presione el botón: Eliminar.
3. Saldrá un mensaje para confirmar que desea eliminar.



4. Haga clic en el botón Aceptar para eliminar o Cancelar si no desea borrar la planta turística del sistema.
5. Haga clic en el botón Cerrar para salir de esta ventana.

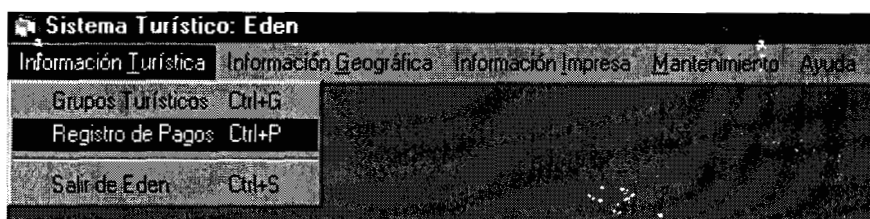
B.6 MANTENIMIENTO DE REGISTRO DE PAGOS

El sistema Edén permite llevar un control de los pagos anuales que realizan las plantas turísticas para su funcionamiento.

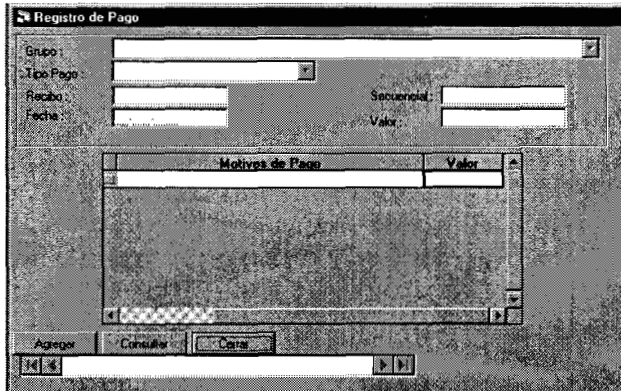
Ingreso de un Registro de Pagos

Haga los siguientes pasos:

1. Escoger del menú principal: *Información Turística/Registro de Pagos*



2. Se mostrará la siguiente pantalla:



3. Presione el botón Agregar.

4. Donde dice Grupo, tiene que escoger la planta turística y luego llenar o seleccionar los demás datos para registrar el pago.

5. Presione el botón Actualizar para grabar o Cancelar para salir sin grabar.

Búsqueda de un Registro de Pagos

Haga los siguientes pasos:

1. Haga los dos primeros pasos de la sección Ingreso de registro de pagos

2. Haga clic en el botón Consultar

3. Escriba el o los datos del registro de pagos que busca y presione el botón Ejecutar. En caso de que no sepa ningún dato, entonces, simplemente presione el botón Ejecutar y verá un listado total de los registros de pago de las plantas turísticas.

4. Luego se mostrará una lista de la cual debe seleccionar el registro haciendo doble clic.
5. Se mostrará todos los datos del registro encontrado.

Modificación de un Registro de Pagos

Haga los siguientes pasos:

1. Haga los dos primeros pasos de la sección Ingreso de registro de pagos
2. Localice el registro que va a modificar
3. Luego cuando se muestre en pantalla los datos, puede modificarlos.
4. Presione el botón Actualizar para grabar o Cancelar para salir sin grabar.

Eliminación de un Registro de Pagos

Haga los siguientes pasos:

1. Haga los dos primeros pasos de la sección Ingreso de registro de pagos
2. Localice el registro que desea eliminar
3. Presione el botón Eliminar.
4. Se mostrará un mensaje para confirmar que desea eliminar el registro.

B.7 MANTENIMIENTO DE LA INFORMACION GEOGRAFICA DE LA PLANTA TURISTICA

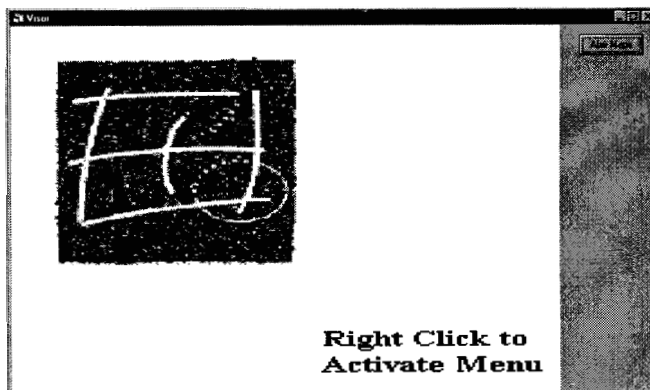
El sistema Edén enlaza la información escrita con la información geográfica de la planta turística. La información geográfica consiste en un mapa de la ubicación física del establecimiento.

Para ver la información geográfica:

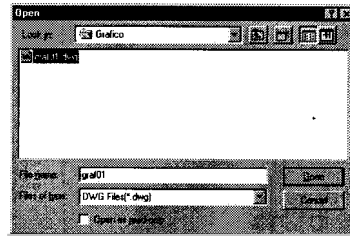
1. Escoja del menú principal: *Información Geográfica / Guayas Ctrl+M*



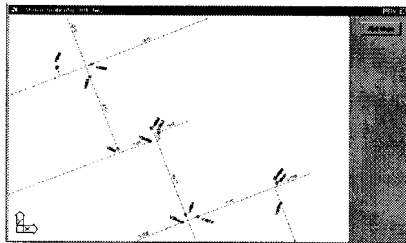
2. Se mostrará la siguiente ventana que es un visor de mapas:



3. Haga clic en el botón: Abrir Mapa
4. Se mostrará una ventana para que seleccione el archivo de mapa a visualizar.

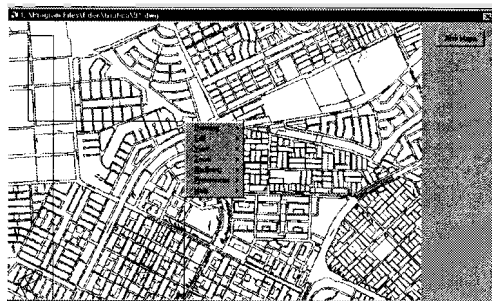


5. Una vez escogido el archivo, presione el botón Open.
6. Luego podrá visualizar el archivo de mapa seleccionado como se muestra a continuación.

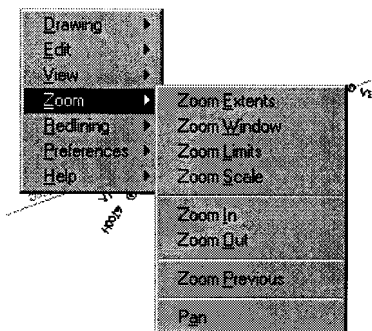


Para ver mejor el mapa utilice el menú Zoom que se explica a continuación:

1. Dentro del área del mapa haga clic con el botón derecho del mouse para que se muestre el menú



2. Del menú Zoom puede seleccionar:



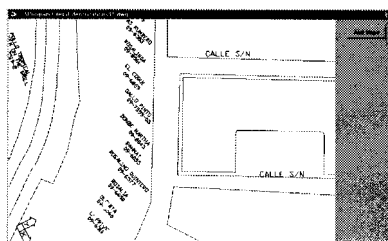
Zoom Extents

Sirve para ver todas las extensiones del mapa, es decir ver el mapa completo.



Zoom Window

Al escoger Zoom Window puedo encerrar en un recuadro la parte del mapa que quiero agrandar más.



Zoom Limits

Como su nombre lo indica, es para ver los límites del mapa.



Zoom Scale

Sirve para poner la escala para **que** se reduzca el tamaño del mapa.

Zoom In

Es para acercarse más al mapa. (Visualizar más grande).

Zoom Out

Es para alejarse más del mapa. (Visualizar más pequeño).

Zoom Previus

Para visualizar con el Zoom anterior.

Pan

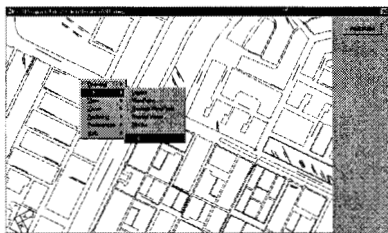
Es para desplazarse en el mapa. Tiene que hacer clic y arrastrar el mouse hacia la dirección que desea que se desplace el mapa. En el gráfico izquierdo la línea que se ve trazada en el mapa es la dirección que se arrastró el mouse. El gráfico de alado es el mapa desplazado.



Copia del Mapa

Para sacar una copia del mapa:

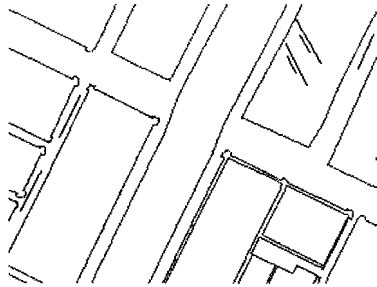
1. Con el botón derecho del mouse seleccione: Edit / Copy



2. El cursor del mouse estará listo para que encierre en un recuadro lo que desea copiar del mapa.



3. Lo que encerró del mapa lo puede copiar en una hoja de Microsoft Word o PowerPoint o Paint.

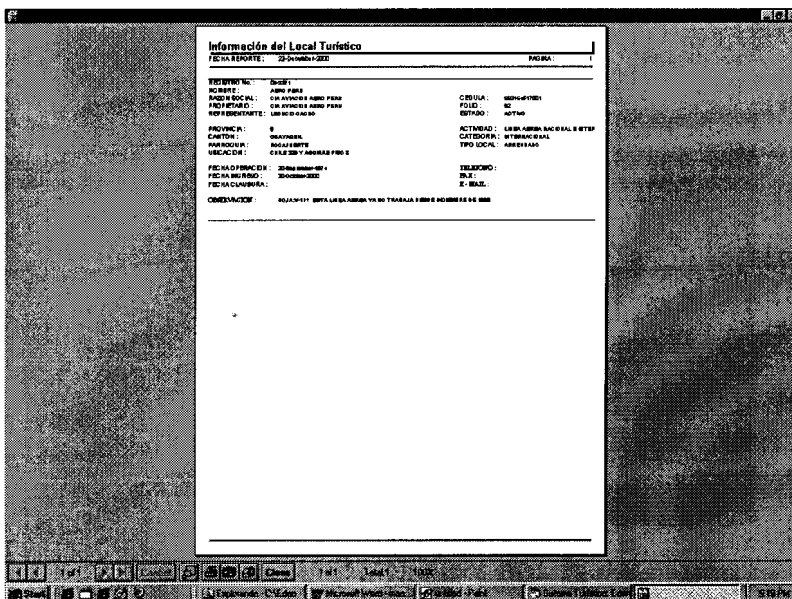


B.8 INFORMACION IMPRESA

Planta Turística (Grupo Turístico)

Imprimir los datos de una planta turística específica:

1. Debe entrar a la ventana: Información Grupo Turístico, y buscar la planta turística
2. Presione el botón: Imprimir y podrá verlo en pantalla. Para mandarlo a imprimir debe hacer clic en el botón que tiene figura de impresora.



The screenshot shows a window titled "Información del Local Turístico" with a "Fecha Reporte" of "23-Diciembre-2002" and a "Página" of "1". The form contains the following data:

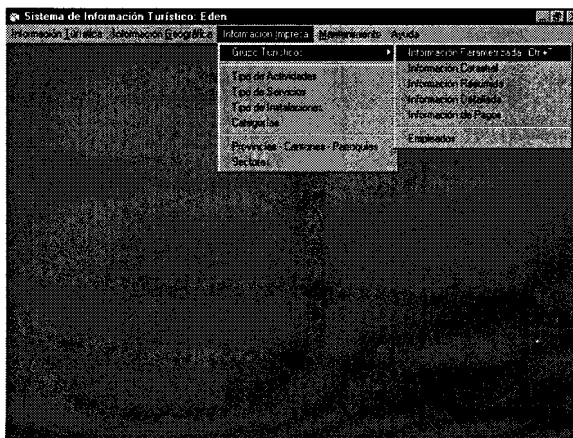
ESTABLECIMIENTO NO.:	50281	CIRCULO:	INDUSTRIAL
ACTIVIDAD:	ALMO PARA	PO. DE:	50
RANGO SOCIAL:	CU. ANONIMO ALMO PARA	ESTADO:	ACTIVO
RANGO ETNIA:	CU. ANONIMO ALMO PARA	ACTIVIDAD:	LEER ALBERG SOCIAL E RSTP
NO. ESTABLECIMIENTO:	4818100000	CATEGORIA:	BT TURISTICAL
PROVINCIA:	9	TPO. SOCIAL:	ALBERGADO
CANTON:	SAQUENSA	TELEFONO:	
MUNICIPIO:	BOGOTATE	TEL.:	
UBICACION:	CALLE 200 Y AVENIDA PNO 2	E. REAL:	
FECHA DE FUNDACION:	2000-00-00		
FECHA DEL REGISTRO:	2000-00-00		
FECHA DE CANCELACION:			
COMENTARIOS:	02/12/02 11:11 BOTA LA UNDA ALBERGADO EN TRABAJO E HUBO PROBLEMA DE SER		

Imprimir los datos de todas las plantas turísticas:

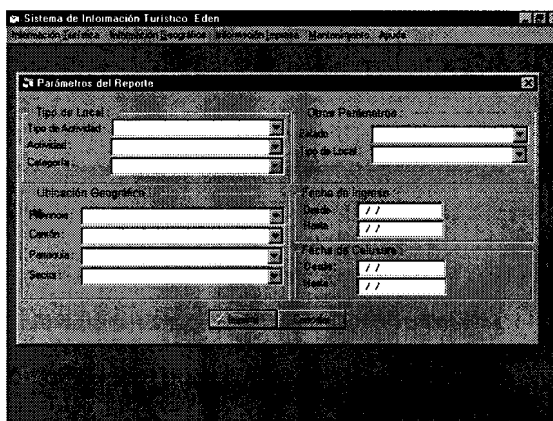
Información parametrizada

Imprime un reporte de los datos generales de las plantas turísticas filtrando la información por cualquiera de los datos mostrados en pantalla.

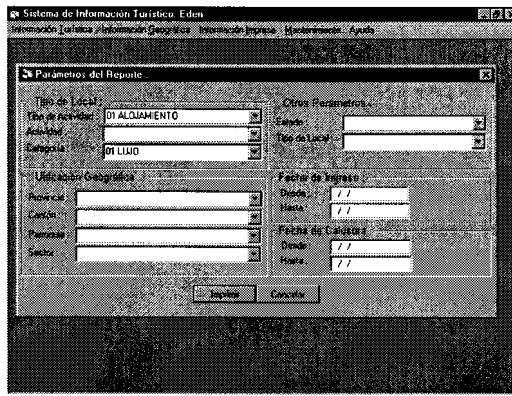
1. Escoja del menú principal: *Información Impresa / Grupos Turísticos / Información Parametrizada*



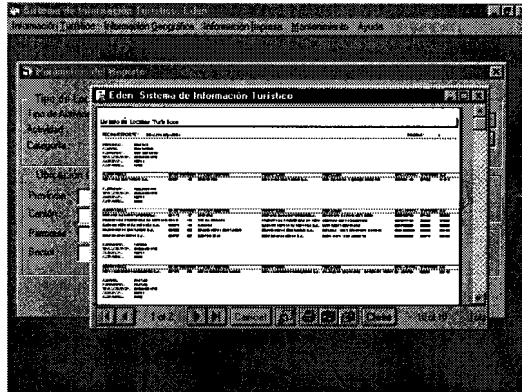
2. Se mostrará la siguiente pantalla:



3. En esta pantalla debe escoger el filtro de la información, es decir, escoger el tipo de actividad, actividad, categoría, etc., no es necesario llenar toda la pantalla de información. Para imprimir debe hacer clic en el botón que dice Imprimir.



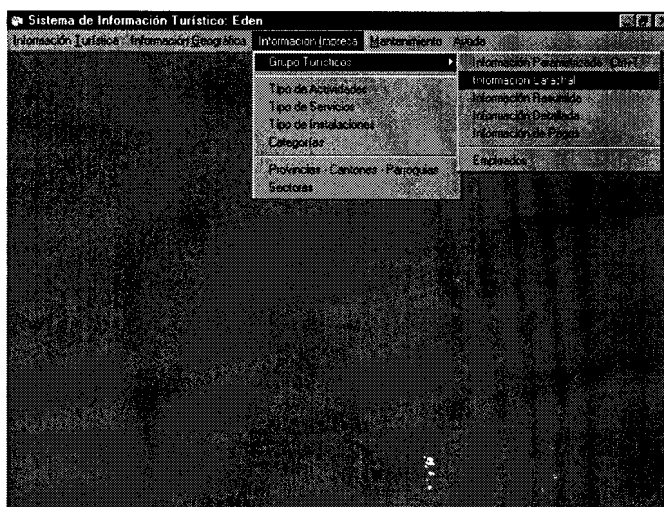
- Se mostrará el reporte a imprimir.



Información catastral

Imprime el catastro.

- Escoja del menú principal: *Información Impresa / Grupos Turísticos / Información Catastral*

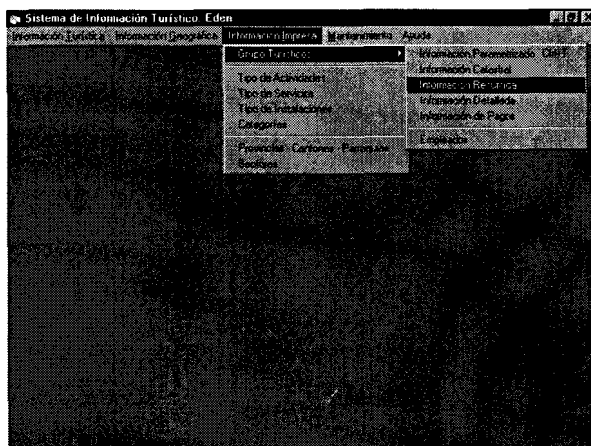


2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Información resumida

Imprime un reporte de los datos generales de todas las plantas turísticas

1. Escoja del menú principal: *Información Impresa / Grupos Turísticos / Información Resumida*

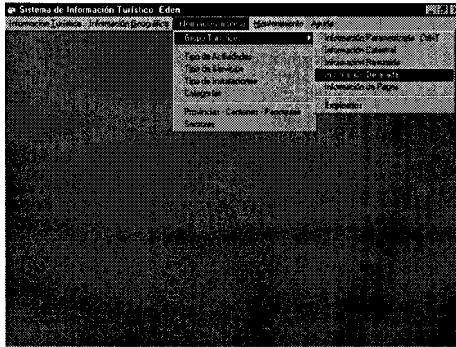


2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Información detallada

Imprime los datos generales y sus detalles de instalaciones de todas las plantas turísticas.

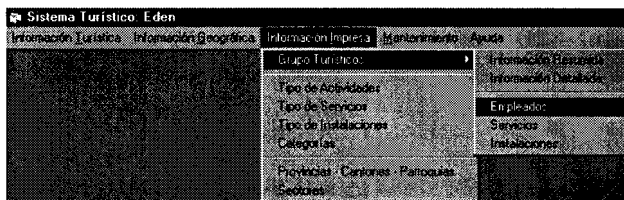
1. Escoja del menú principal: *Información Impresa / Grupos Turísticos / Información Detallada*



2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

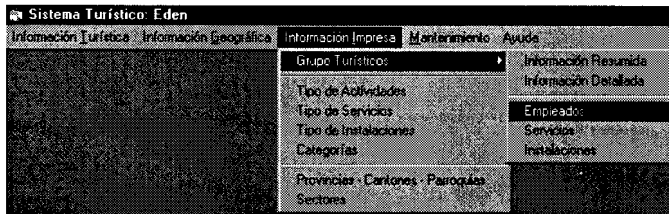
Información de pagos

1. Escoja del menú principal: *Información Impresa / Grupos Turísticos / Información de pagos*



Empleados

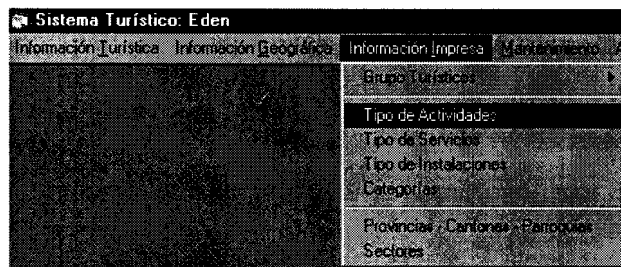
1. Escoja del menú principal: *Información Impresa / Grupos Turísticos / Empleados*



3. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Tipo de Actividades

1. Escoja del menú principal: *Información Impresa / Tipo de Actividades*



2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Tipo de Servicios

1. Escoja del menú principal: *Información Impresa / Tipo de Servicios*
2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Tipo de Instalaciones

1. Escoja del menú principal: *Información Impresa / Tipo de Instalaciones*
2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que *se* mostrará en la parte inferior del reporte.

Categorías

1. Escoja del menú principal: *Información Impresa / Categorías*
2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Provincias – Cantones - Parroquias

1. Escoja del menú principal: *Información Impresa / Provincias – Cantones - Parroquias*

2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

Sectores

1. Escoja del menú principal: *Información Impresa /Sectores*
2. Automáticamente se mostrará en pantalla el reporte. Para imprimir debe hacer clic en el botón de impresora que se mostrará en la parte inferior del reporte.

ANEXO C

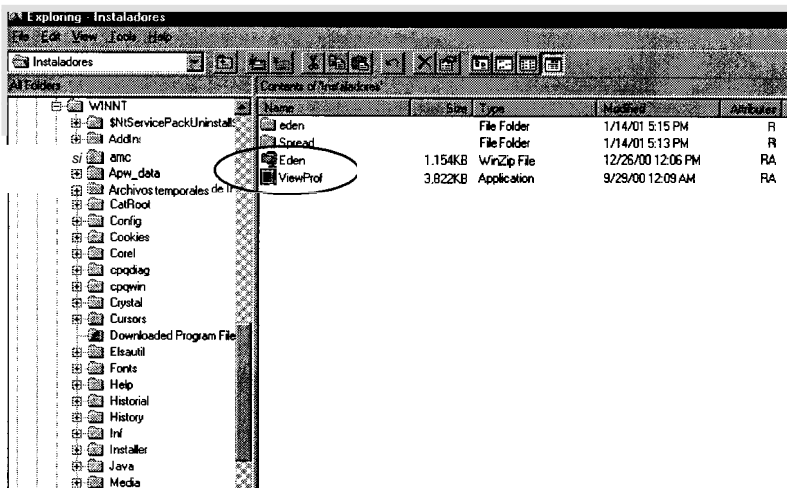
*Manual de Instalación del
Sistema de Información
Turístico (SIT) Eden*

C.1 INSTALACION DEL SISTEMA

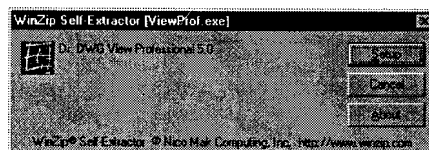
Antes de instalar el Sistema Edén, debe instalar Dr. View Professional

Para instalar Dr. View Professional:

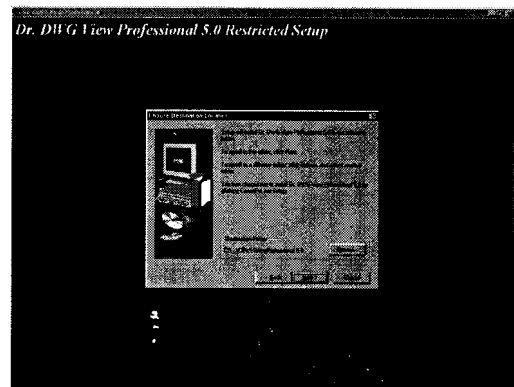
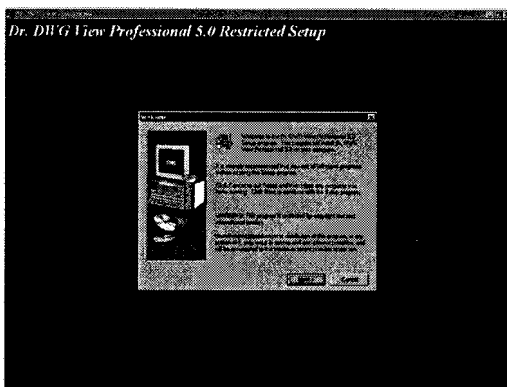
1. Busque en el explorador de Windows el archivo ViewProf.exe



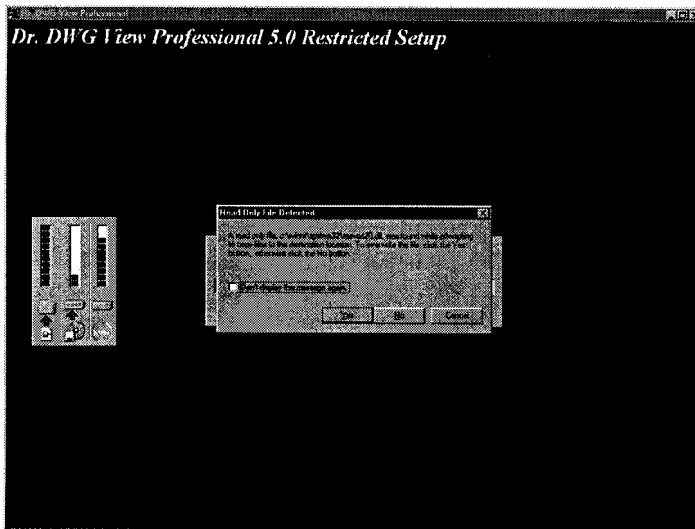
2. Haga doble clic sobre el icono del archivo.
3. Se mostrará una ventanita en la cual debe presionar el botón Setup



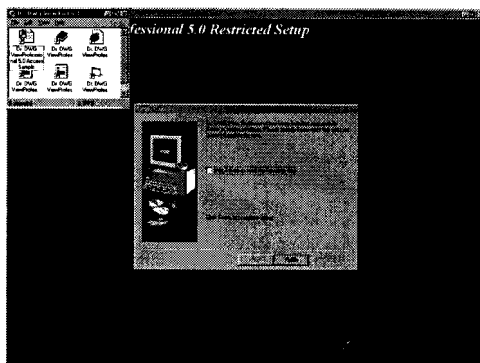
4. Luego se presentarán algunas ventanas en la cual debe presionar el botón Next



5. Luego se mostrará un mensaje en el cual debe presionar el botón: No

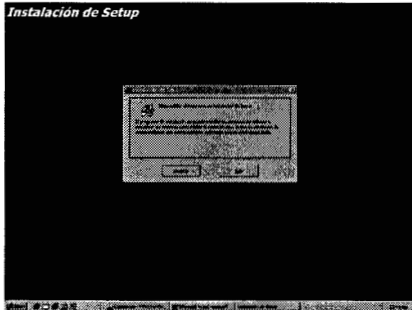


6. Luego al terminar la instalación se mostrará la siguiente ventana e la cual debe hacer clic en el botón Finish

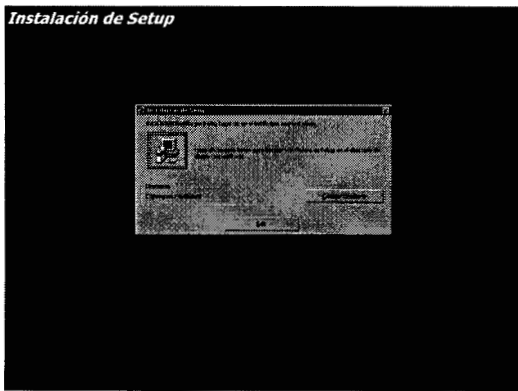


Para instalar el sistema:

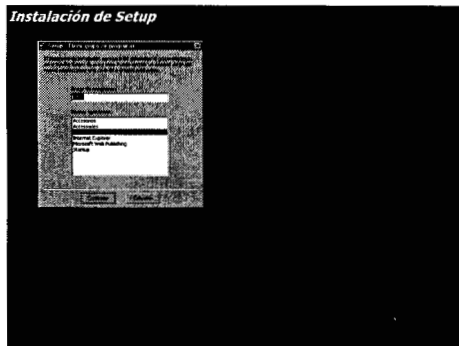
1. Ponga el diskette de instalación y busque el archivo setup
2. Comenzará a correr el programa de instalación del sistema Edén.
3. Presione el botón Aceptar



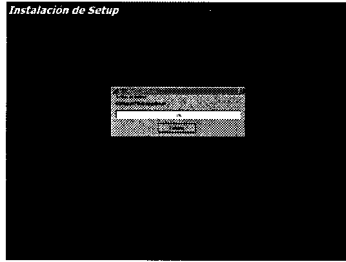
4. Luego presione el botón que tiene figura de computadora



5. Presione el botón Continuar



6. Luego automáticamente continuará la instalación del sistema



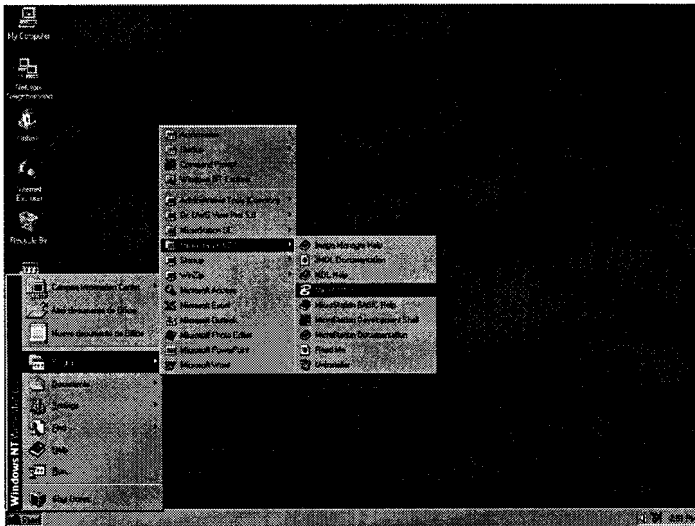
7. Luego que haya terminado la instalación podrá correr el sistema Edén desde la barra de Inicio en menú Programas y busque Edén.

ANEXO D

*Enlaces del Mapa a la base
de datos del Sistema de
Información Turístico (SIT)
Eden*

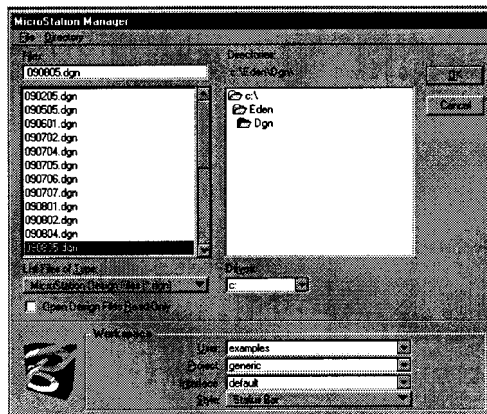
D.1 Enlace del mapa a la base de datos Eden

1. Debe ingresar a MicroStation



2. Se mostrará una ventana en la cual debe presionar el botón OK

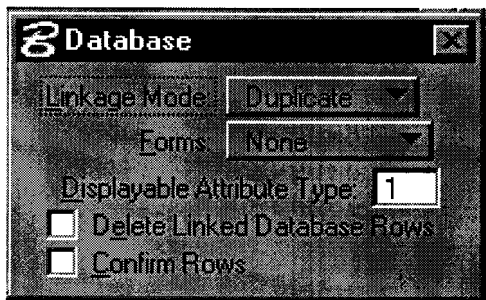
3. Luego debe escoger el archivo dgn que va a abrir



4. Ir al menú: Settings/ Database / Dialog



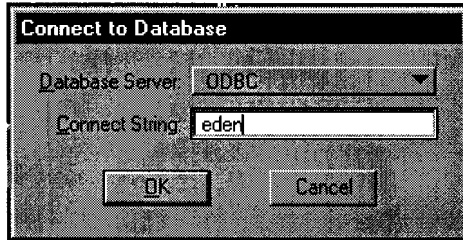
5. Luego se mostrará una ventanita en la cual debe escoger: Duplicate



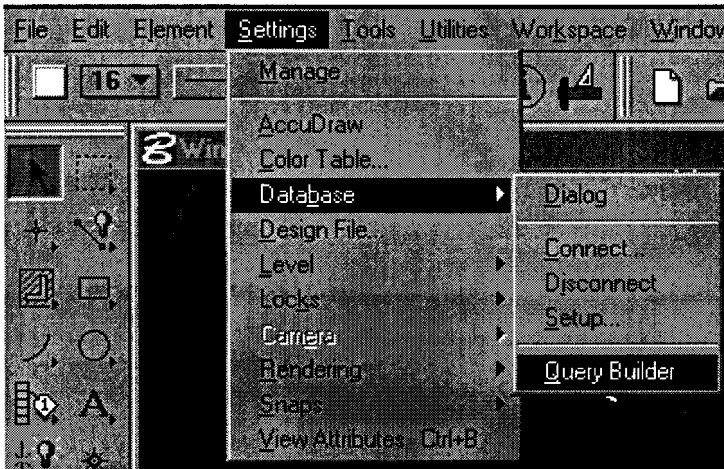
6. Cerrar esa ventanita y escoger del menú: Settings/ Database / Connect



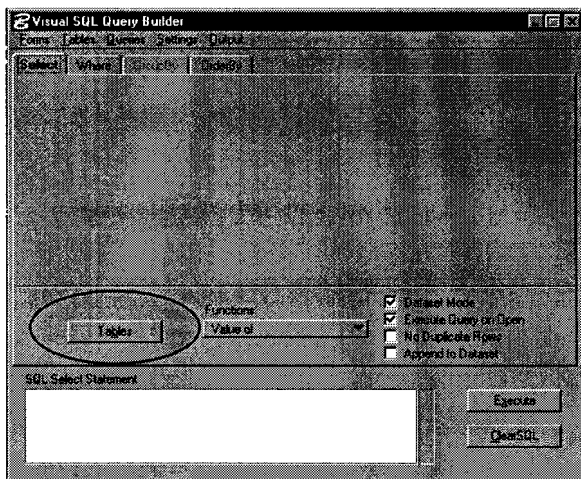
7. Se mostrará una ventana en la cual debe escoger ODBC y escribir el nombre de la base, en este caso, Eden y presionar el botón OK



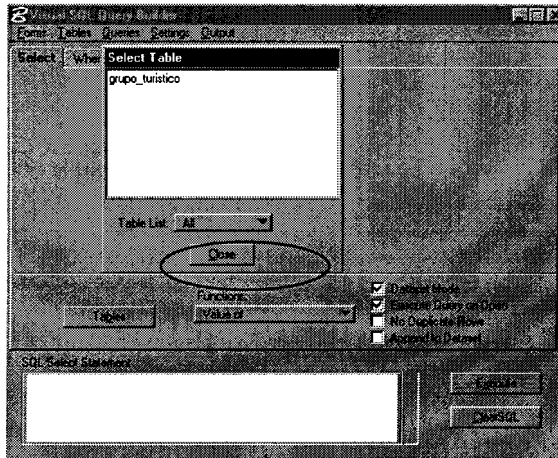
8. Ir al menú: Settings/ Database/ Query Builder



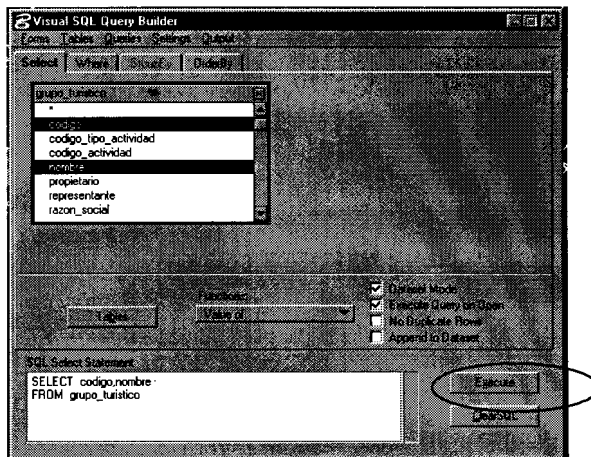
9. Se mostrará una ventana en la cual debe presionar el botón: Table



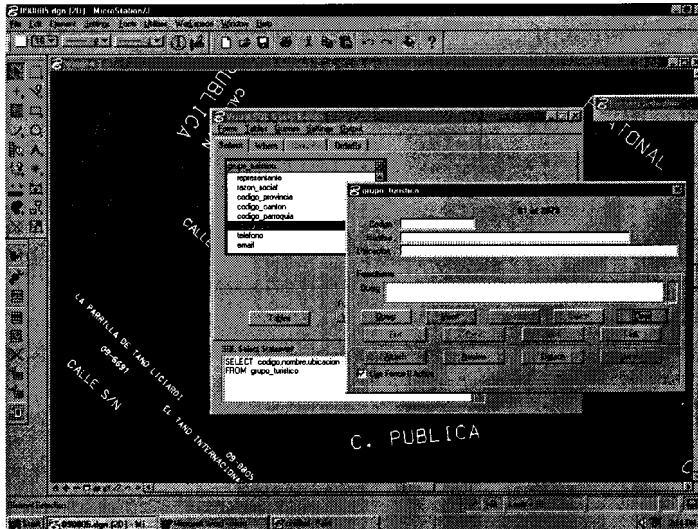
10. Luego debe escoger el nombre de la tabla, en este caso, grupo_turistico, haciendo doble clic. Luego hacer clic en el botón Close.



11. Escoja los campos que quiere ver para hacer el enlace. (código, nombre y ubicación). Presionar el botón Execute

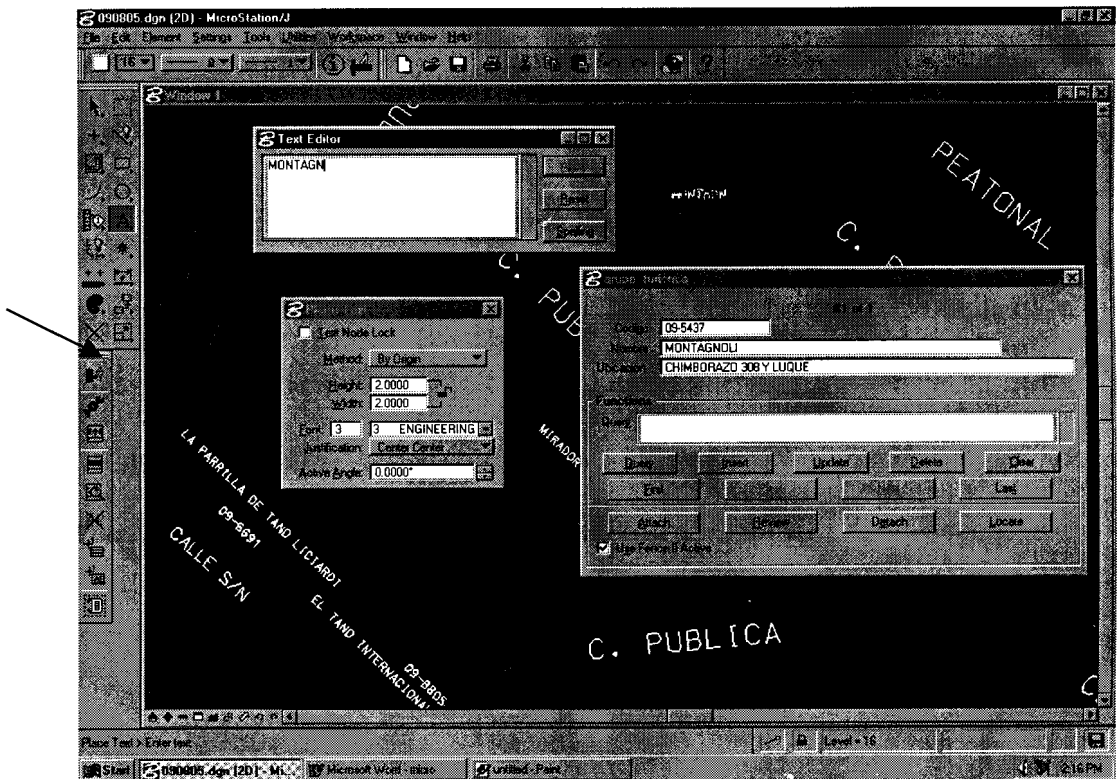


12. Se mostrará la siguiente ventana

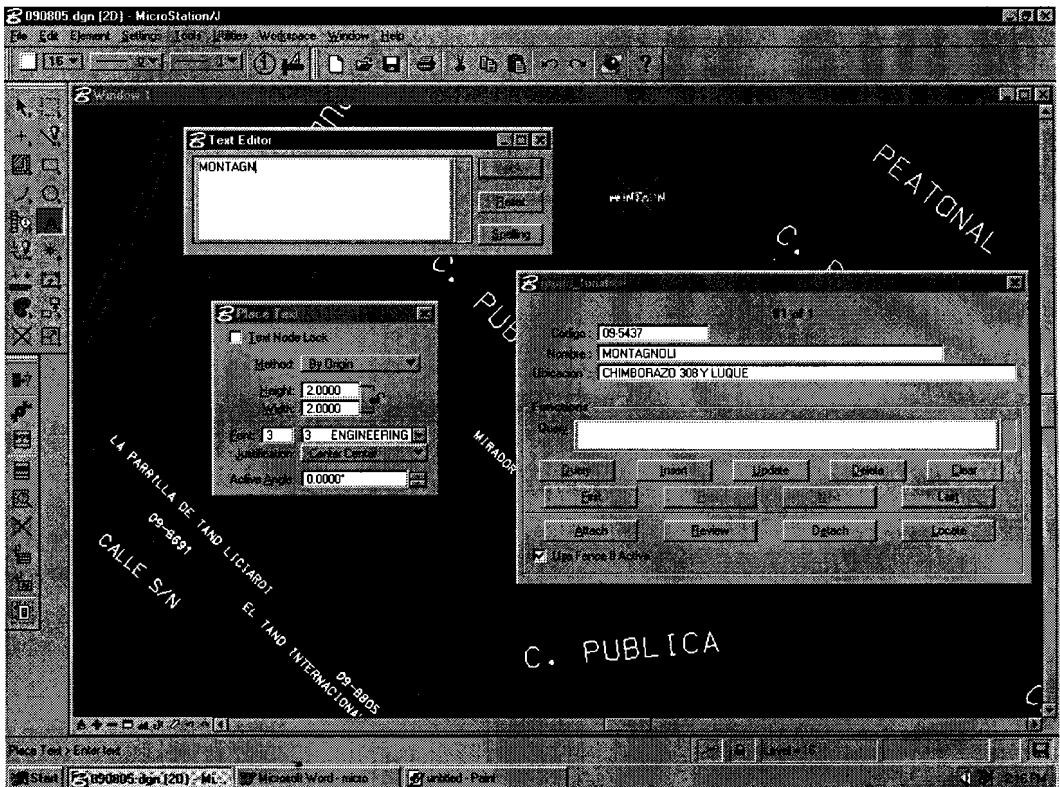


13. En la caja de texto código, escriba el código del establecimiento y luego presione el botón Query. Se llenará la información del establecimiento.

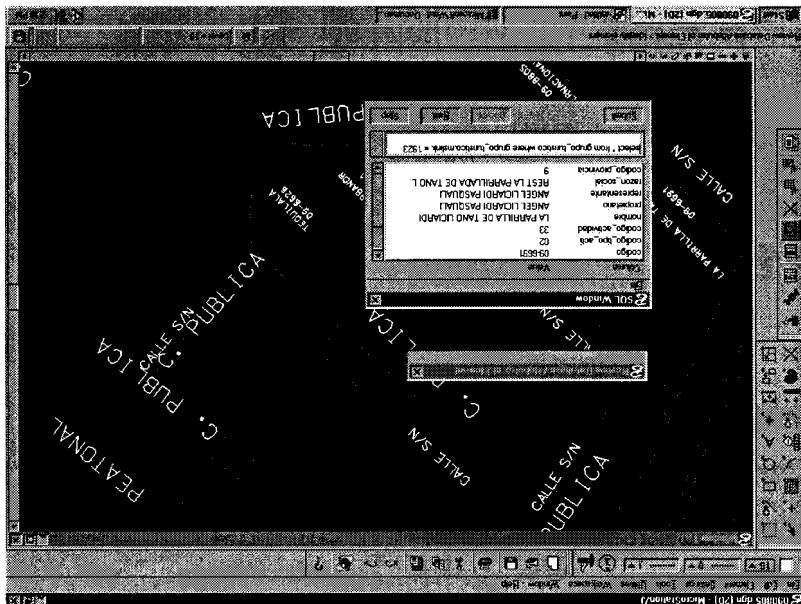
14. Presione en la barra de herramientas el botón de texto. Botón A



15. Escriba en la caja de texto el nombre del establecimiento y ubíquelo en el mapa.
16. Luego escriba el código del establecimiento.
17. Presione el botón Attach



18. Para verificar que se hizo el enlace habilite la barra de herramientas de la base de datos escogiendo del menú Tools / Database. Luego, en la barra de herramientas Database escoja el icono "Review Database Attributes of Element" y haga doble clic sobre el código. Se mostraron los datos del código.



ANEXO E



Manual de Georeferenciación y Digitalización



E.1 REQUERIMIENTOS DEL SISTEMA DE HARDWARE Y SOFTWARE

HARDWARE

- Computador con las siguientes características:
 - Procesador Intel Pentium actualizable a Pentium III
 - Disco Duro de 2GB mínimo expandible a 4GB en adelante
 - Memoria RAM de 64MB mínimo expandible a 128MB
 - Memoria de Video de 4MB en adelante
 - Monitor de 17" SVGA con alta resolución

SOFTWARE

- Requerimientos de Software:
 - Sistema Operativo Windows NT en idioma Español ó idioma Inglés para el Servidor
 - Sistema Operativo Windows NT WorkStation en idioma Inglés ó Windows 95 en Idioma Inglés para el Cliente
- Sistema de Información Geográfico MicroStation 95
- RASB, IGEOVEC como herramientas de digitalización sobre rasters (archivos escaneados)
- MGE como herramienta de enlace a la Base de Datos

E.2 PROCESO DE ESCANEEO

Para la digitalización de la cartografía existen dos formas de realizarla, la primera consiste en **utilizar mesas digitalizadoras** que retroalimentan directamente desde el mapa original al sistema, y la segunda es **utilizando rasters** que corresponde a los archivos que se crean al **escanear** los mapas usando formato E y/o **Ao**.

Para el mencionado proyecto se ha escogido la **segunda opción**, se clasifican los mapas de acuerdo a la resolución que hay que dar al momento de escanear para que la imagen sea nítida y no haya ningún inconveniente al momento de la digitalización sobre ellos, en los parámetros que hay que ingresar en el software de manejo del escáner se escoge el formato AO, Resolución de 800 DPI y un contraste base de 80-100 que puede variar de acuerdo a la nitidez de la hoja que se esté escaneando.

E.3 ARCHIVOS RASTER Y GEOREFERENCIACION

Una vez que los mapas han sido escaneados, se convierten en archivos rasters cuya extensión es **CAL**, para proceder a la digitalización sobre ellos se debe en primer lugar trasladar o georeferenciar a las coordenadas exactas en el espacio a donde pertenece cada ciudad de acuerdo al sistema de coordenadas establecido por el Instituto Geográfico Militar para la Provincia del Guayas.

Cuando los archivos rasters han sido georeferenciados, se deben guardar con el nombre respectivo del mapa al que pertenece creándose de ésta manera archivos con extensión **.CIT**

Para el efecto de manejar archivos rasters y proceder con su respectiva georeferenciación se debe utilizar una de las herramientas del Sistema de Información Geográfica denominada **IGEOVEC**.

Terminado el proceso de georeferenciación se prosigue con la digitalización sobre los archivos **.CIT** utilizando la misma herramienta ó las herramientas brindadas por **MICROSTATION95/98**

E.4 PASOS A SEGUIR PARA LA GEOREFERENCIACION

- a. Se debe ejecutar el programa **IGEOVEC**, en el cual se presentarán tanto el menú de **MICROSTATON** y **IGEOVEC-IRASB**

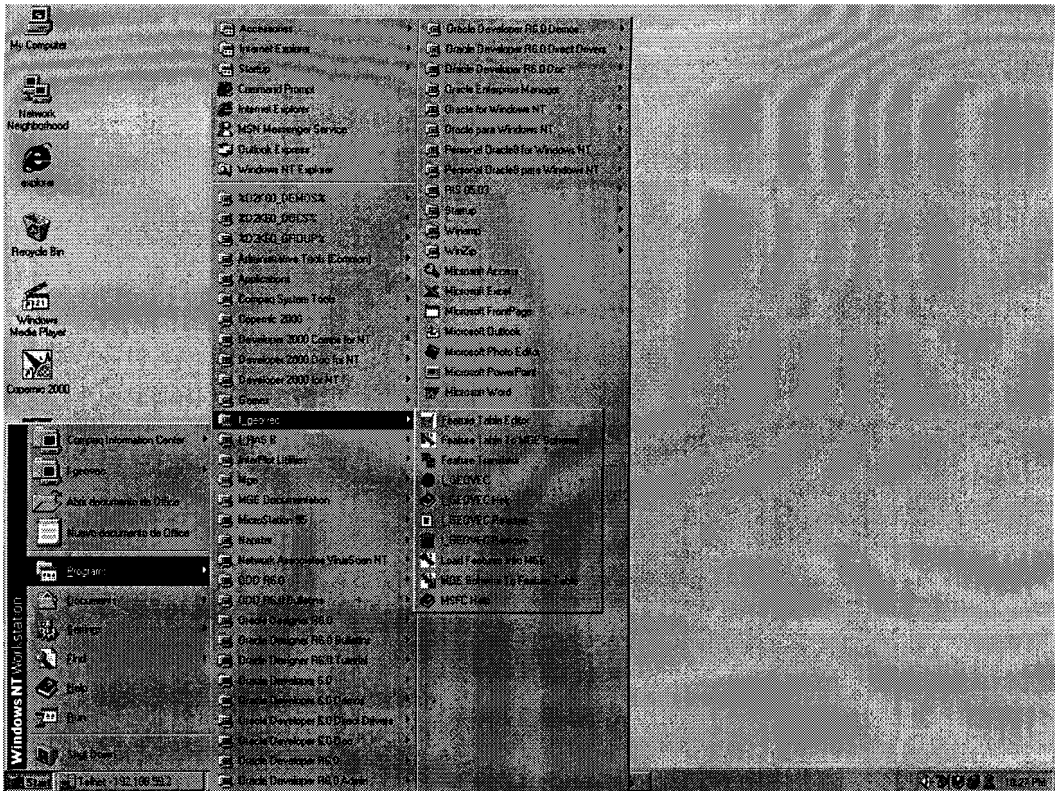
Cuando los archivos rasters han sido georeferenciados, se deben guardar con el nombre respectivo del mapa al que pertenece creándose de ésta manera archivos con extensión **.CIT**

Para el efecto de manejar archivos rasters y proceder con su respectiva georeferenciación se debe utilizar una de las herramientas del Sistema de Información Geográfica denominada **IGEOVEC**.

Terminado el proceso de georeferenciación se prosigue con la digitalización sobre los archivos **.CIT** utilizando la misma herramienta ó las herramientas brindadas por **MICROSTATION95/98**

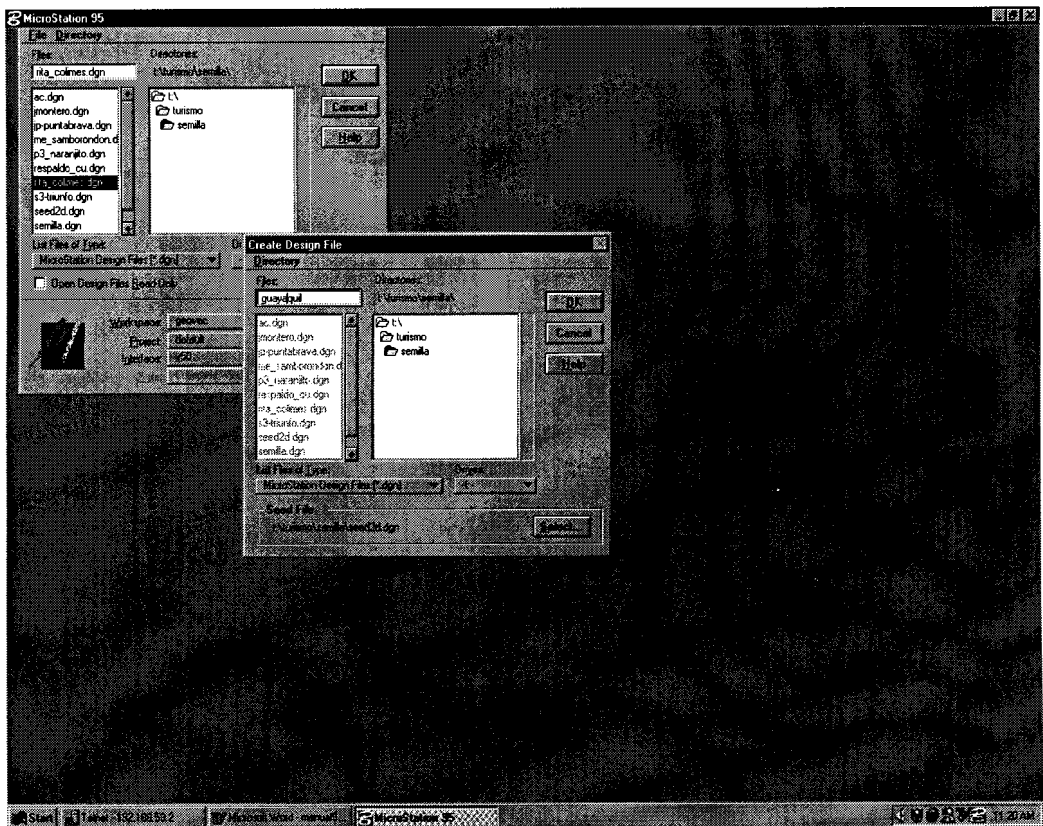
E.4 PASOS A SEGUIR PARA LA GEOREFERENCIACION

- a. Se debe ejecutar el programa **IGEOVEC**, en el cual se presentarán tanto el menú de **MICROSTATON** y **IGEOVEC-IRASB**



b. Antes del proceso de georeferenciación se debe crear un nuevo archivo extensión **.DGN** en base a un archivo semilla de dos dimensiones que se encuentra en el directorio:

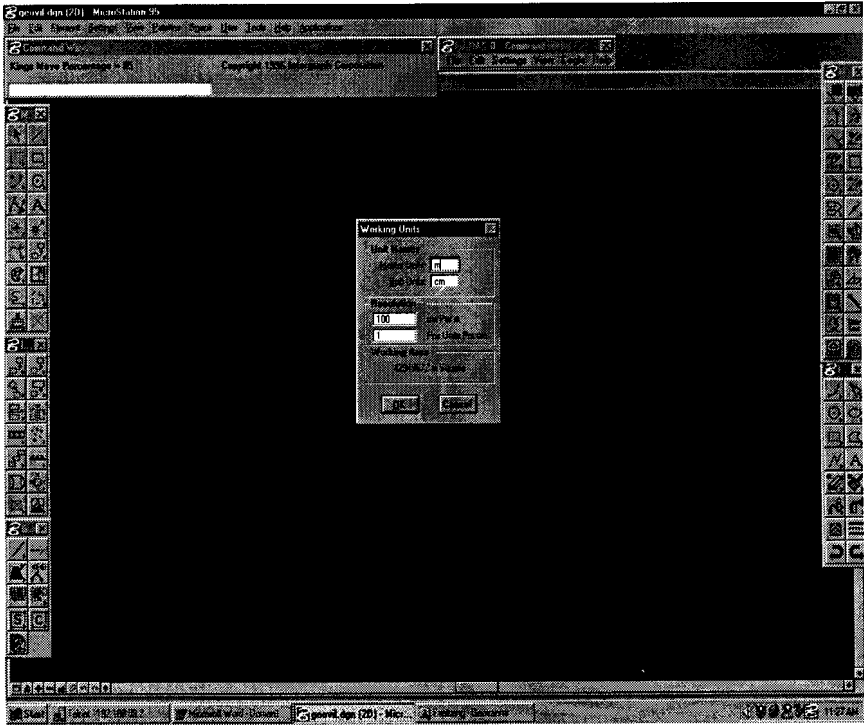
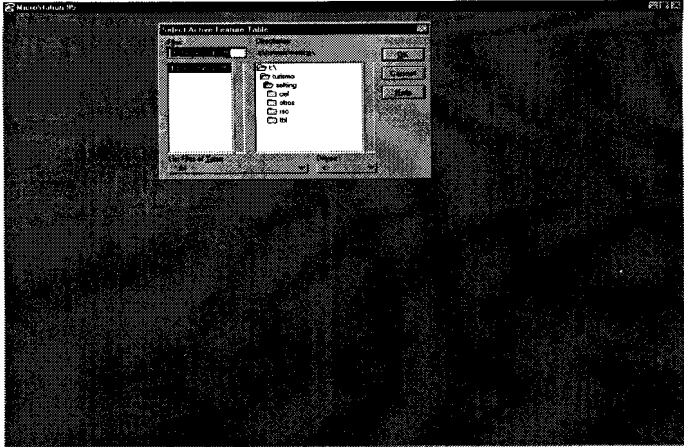
Win32app-Ustation-Wsmod-default-seed-seed2.dgn



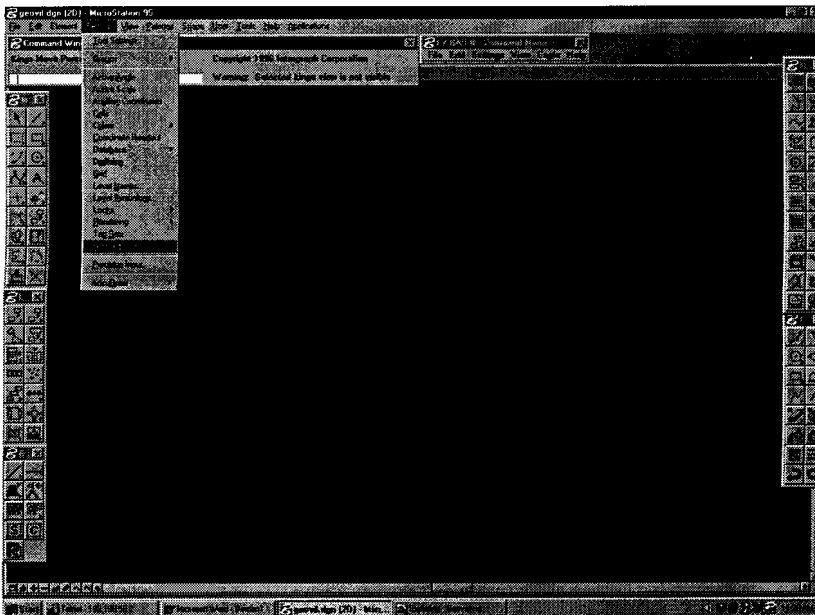
- c. El nuevo archivo debe guardarse con el nombre de acuerdo al mapa sobre el cual se vaya a digitalizar en ese momento y que corresponde al archivo raster que ha sido escaneado.

- d. Luego de crear el archivo, pedirá las tabla de características con la que debe trabajar, existe un feature por default denominado “**features.tbl**” pero para efectos del proyecto se ha creado un feature específico que contiene todas las características bases para la digitalización de los mapas, denominado:

turismo.tbl



- e. Terminado esto se ingresa ya a la hoja de trabajo de MicroStation y IRASB, en la barra de menú de MicroStation se debe elegir la opción Settings y dentro de ella Working Units para setear las unidad de medida en la que se va a trabajar, para el proyecto se ha establecido que las unidades serán metro-centímetro, se debe colocar de la siguiente manera:

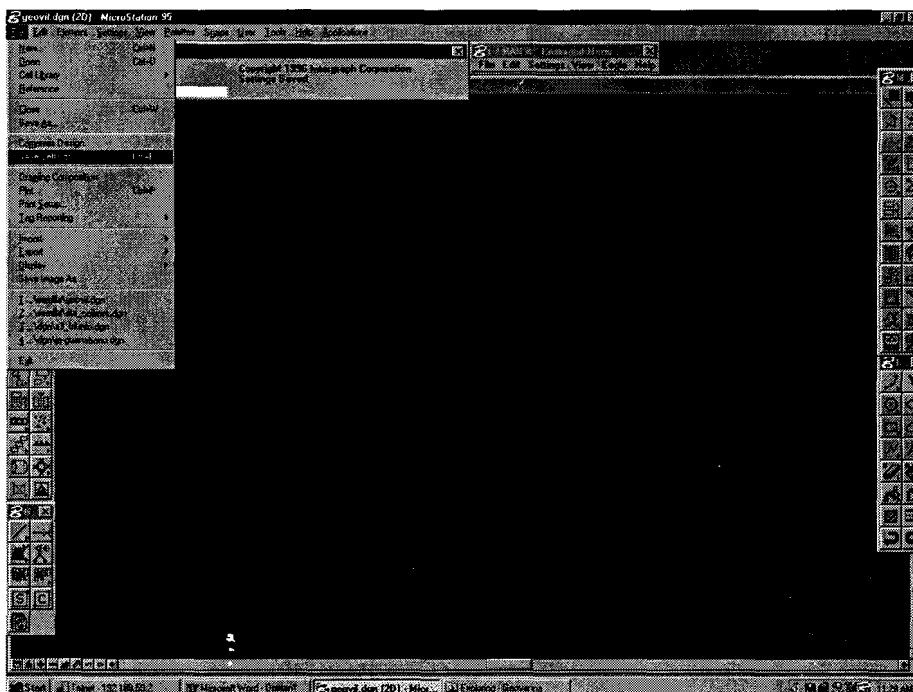


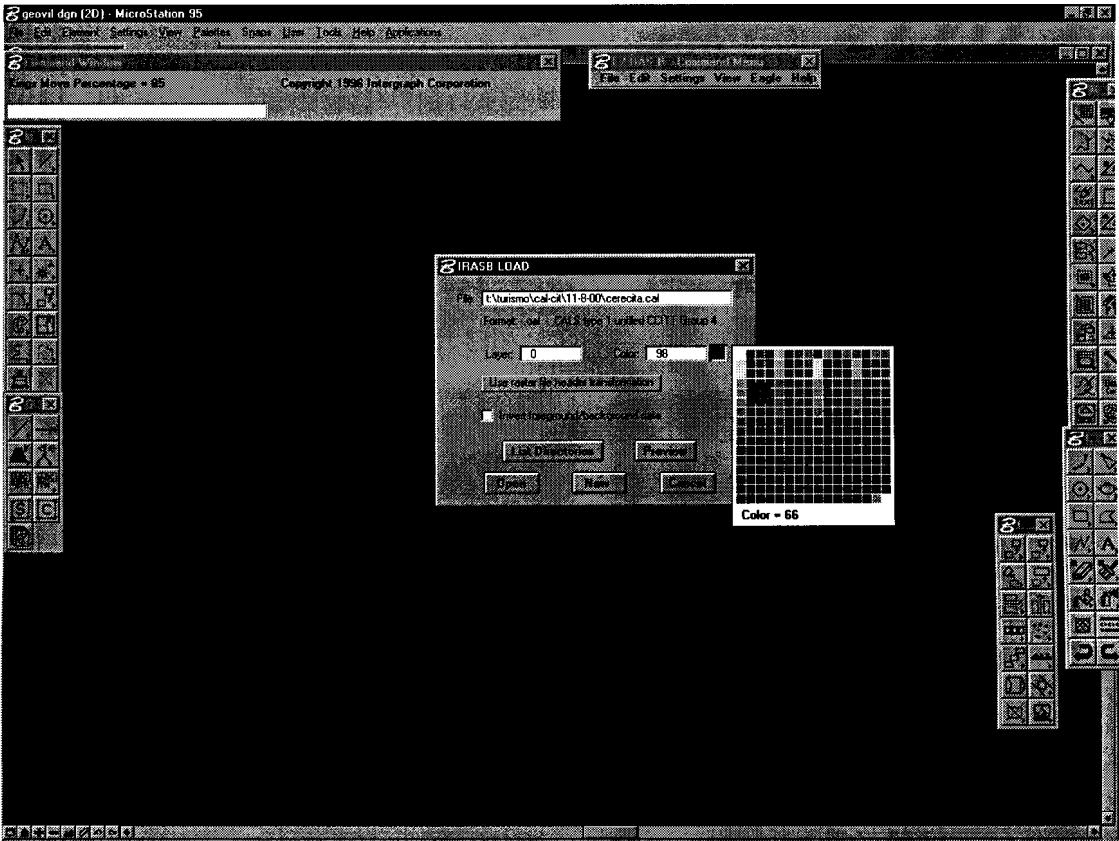
- f. Configurado el literal e, ir al menú file y escoger la opción **SAVE SETTINGS** (CTRL+F), ésta opción siempre debe ser utilizada cuando se hagan cambios en algún seteo de las opciones de MicroStation

- g. Entonces se procede a dibujar un rectángulo cuyas esquinas superior izquierda e inferior derecha formen el área dada por las coordenadas correspondientes al mapa de cada una de las provincias del Guayas.

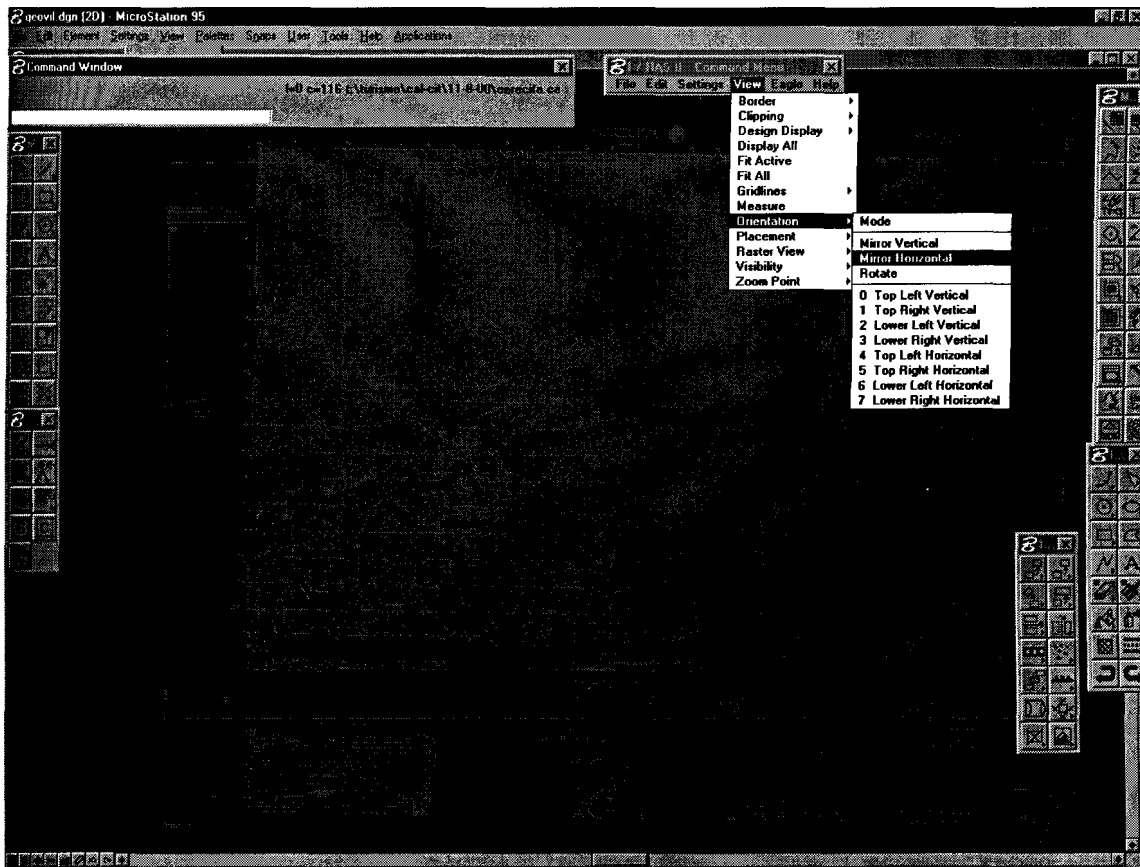
Para trazar el rectángulo se debe hacer utilización de una de las herramientas de MicroStation denominada **Place Block** en donde hay que ingresar en la caja de texto el valor de las dos coordenadas planas principales que definen el rectángulo en los puntos XY para la correcta georeferenciación.

- h. Una vez formado el rectángulo, hacer uso del menú de IRAS/B para llamar al raster correspondiente cuya extensión es **.cal** y deben tener el mismo nombre tanto raster como archivo **dgn** para efectos de asociación y organización, se debe elegir un color visible al usuario para trabajar con el raster

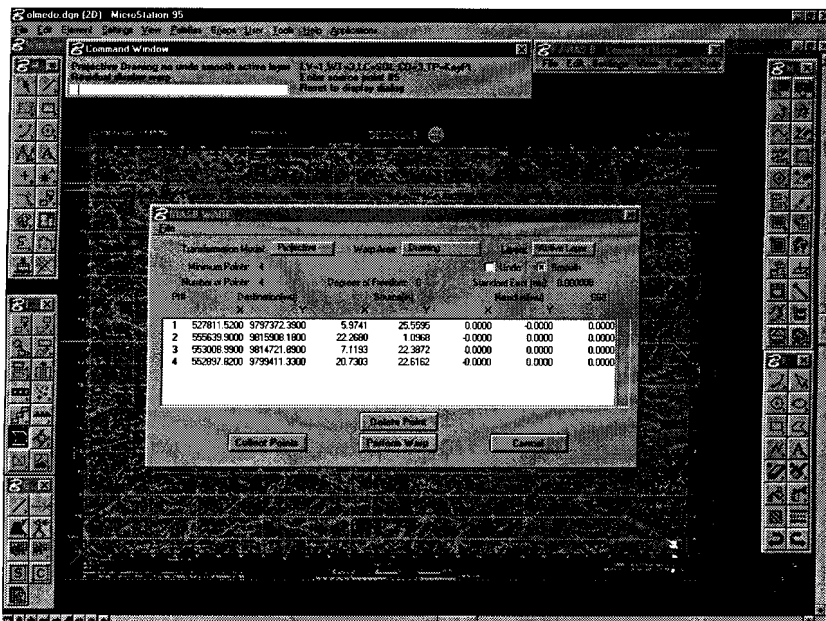
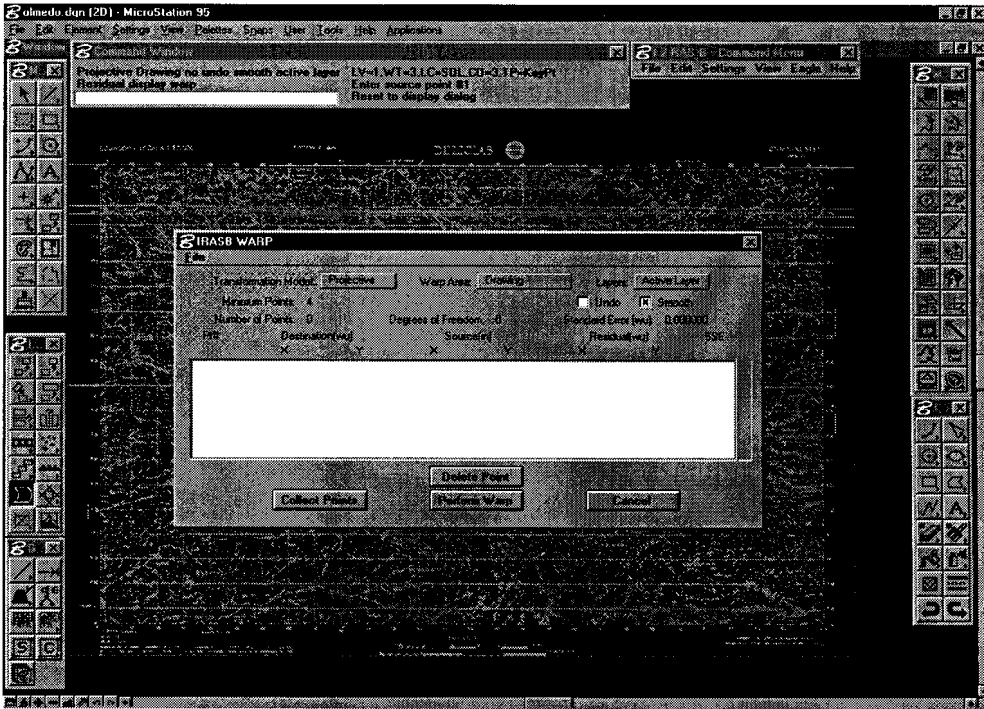


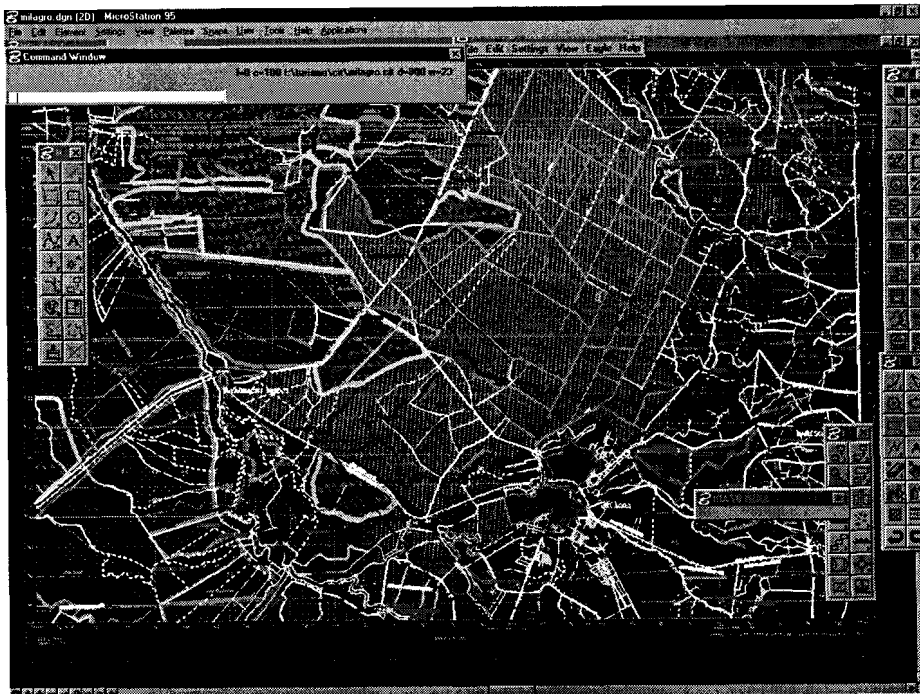


- i. Muchas veces cuando se trae el archivo raster en ocasiones habrá que rotarlo utilizando la opción Orientation - Mirror Horizontal, Mirror Vertical del menú View de la barra de IRASB y grabar los cambios en el raster.
- j. Cerrar el archivo y abrirlo nuevamente tratando de que quede ubicado en la misma posición del rectángulo, una vez logrado, proceder a la georeferenciación del raster.



- k. En la barra de menú de IRASB seleccionar **Edit** y la opción **Modify-Warp-Dialog**. Se visualiza una ventana cuya funcionalidad es para colectar puntos, se debe clicar sobre el comando **Collection Points**, escoger para nuestro caso **4** coordenadas alternando entre archivo **.Cal** y archivo **.dgn** que en un principio es el rectángulo previamente digitalizado y con respecto al cual lo vamos a trasladar, luego proceder a realizar click sobre el comando **Perform Warp** para lograr la georeferenciación.





1. Terminado de ejecutar, proceder a verificar que el archivo raster esté correctamente georeferenciado, es decir que los puntos de coordenadas que fueron tomados coincidan exactamente con los vértices del rectángulo del archivo dgn.

- m. Si todo coincide perfectamente grabar éste nuevo archivo con conservando el mismo nombre, ir a menú **File** de IRASB, escoger la opción **Save-Active Layer As** y en el campo **Format** escoger la extensión **.cit**

E.5 NIVELES DE INFORMACION

Dentro de MicroStation existe una opción Levels dentro del menú View que me sirve para tener una mejor guía de que información digitalizada se encuentra guardada en determinado nivel.



Cuando uno de los niveles está seleccionado con un círculo significa que está activo y que lo que se encuentra digitalizando en ese momento se está guardando en esa capa.

Los seleccionados con un cuadrado negro significa que las capas en ese estado están prendidas, toda la información guardada en esos niveles están siendo visibles al usuario, si la selección es gris la información guardada en ese nivel se encuentra oculta hasta ser activada nuevamente.

A continuación se muestra una tabla que contiene la información guardada en cada nivel con sus respectivas características:

NOMBRE	DETALLE	NIVEL	COLOR	PESO	ESTILO
TEXTO	Poblado 2 y límites	31	0		CAPS,112
	Poblados Dispersos(según extensión del elemento)	35	6		C/L,115
	Fincas, haciendas	15	126		C/L,119
	Edificios Públicos, objetos localizados	19	5		C/L,119
	Escuelas, colegios, jardín de infantes	8	5		C/L,119
	Iglesias	20	5		C/L,119
	Cota más alta	28	3		110
	Barrios y suburbios	18	6		C/L,110
	Parques nacionales, reservas ecológicas, etc.	55	2		CAPS,110
	Valores de Cota	11	0		110
	Puntos de control, valor	14	4		110
	Puntos de control, nombre	39	4		GIL,120
	Relieve, pampas, lomas y cerros grandes	40	6		CAPS,117
	Descripción tipo de terreno(lava, rocas,arena,nubes)	12	6		C/L,120
	Vías-ferrocarriles-carretera	26	0		CAPS,120
	Áreas limitadas(polígonos,canteras,cavernas)	13	9		C/L,116

	Valores de curvas de nivel	10	4		116
	Descripción de elementos negros(gasoducto,oleducto,estadios,muros,dique e	9			C/L, 120
	Fondeadero, área peligrosa, etc.(Hidrografía costanera)	37	23		C/L, 120
RIOS	Sencillo 4-8 cm	4	7		
	Sencillo 8-16 cm y camaroneras con nombre	5	7		
	Sencillo más de 16 cm de largo	6	7		
	Océanos, mares, lagos(según extensión del elemento)	7	1		CAPS, 12
FERROCARRI LES	Puente de ferrocaml	41	5	2	0
	Paso elevado	43	5	2	0
	Túnel de ferrocarril	44	6	3	0
	Vía normal	1	0	1	0
	Vía estrecha	3	9	2	0
	Estación ferroviaria	2	6	0	0
VIAL	Autopista con parterre	1	3	3	0
	Paso elevado autopistas	50	5	2	0
	Pavimentada superficie dura 2 vías(camino 1)	2	3	1	0
	Pavimentada superficie dura 1 vía	3	3	1	0
	Afirmado 2 vías	4	4	2	3
	Afirmado 1 vía	5	3	1	2
	Camino de herradura	8	0	1	3
	Camino de tierra o de verano (camino 5)	7	116	1	0

	Sendero o vereda(7)	9	3	2	
	Puentes para vehículos	41	5	2)
	Paso elevado peatonal	57	5	2)
	Carretera en construcción clasificación desconocida	12	6	1)
	Vía planificada(proyecto)	13	12	1	5
	Calles en arreas construida	16	3	1	3
	Puentes peatonales	44	4	2	3
	Túnel vehículos	47	4	1	3
	Intercambiador	6	3	0	0
VEGETACION	Límite de vegetación	1	2	0	0
	Texto de vegetación	2	2	0	0
INF. CUADRICULA	Línea negra(cuadrícula principal , línea gruesa)	19	4	2	0
	Línea negra(cuadrícula principal , línea delgada)	2	4	1	0
	Texto cuadrícula principal números pequeños	10	5	0	0
	Segmentos de minuto	1	0	0	0
	Proyección	1	0	0	0
	Valores geográficos y minutos cuadrícula principal	15	4	0	0
DRENAJE	Area de inundación natural	52	4	5	0
	Arrozal	49	7	5	0
	Manglar	53	2	5	0
	Límite de drenaje(ríos dobles, límite de costas, islas, etc.)	60	1	2	0

	Valores de curvas de nivel	10	4		116
	Descripción de elementos negros(gasoducto,oleducto,estadios,muros,dique e	9			C/L, 120
	Fondeadero, área peligrosa, etc.(Hidrografía costanera)	37	23		C/L, 120
RIOS	Sencillo 4-8 cm	4	7		
	Sencillo 8-16 cm y camaroneras con nombre	5	7		
	Sencillo más de 16 cm de largo	6	7		
	Océanos, mares, lagos(según extensión del elemento)	7	1		CAPS, 12
FERROCARRI LES	Puente de ferrocaml	41	5	2	0
	Paso elevado	43	5	2	0
	Túnel de ferrocarril	44	6	3	0
	Vía normal	1	0	1	0
	Vía estrecha	3	9	2	0
	Estación ferroviaria	2	6	0	0
VIAL	Autopista con parterre	1	3	3	0
	Paso elevado autopistas	50	5	2	0
	Pavimentada superficie dura 2 vías(camino 1)	2	3	1	0
	Pavimentada superficie dura 1 vía	3	3	1	0
	Afirmado 2 vías	4	4	2	3
	Afirmado 1 vía	5	3	1	2
	Camino de herradura	8	0	1	3
	Camino de tierra o de verano (camino 5)	7	116	1	0

	Lago intermitente línea	21	7	2	2
	Embalse, reservorio (línea)	10	7	1	0
	Río perenne	6	1	2	0
	Río intermitente	19	7	2	3
	Cataratas pequeña, río sencillo	10	10	1	
	Canal	27	47	2	0
	Acequia	31	12	1	0
	Acueducto elevado	35	47	1	7
	Acueducto subterráneo, tubería subterráneo	36	47		1
	Acueducto por túnel	37	47		4
	Area de salina, evaporador de sal	24	1		0
	Camaronera, vivero de peces (línea)	23	7		0
	Salitral	16	47		3
	Pozo perenne	48	1		0
	Cisterna	46	7	0	0
	Manantial permanente	13	1	0	0
	Flecha de dirección de corriente	15	5		0
ALTIMETRIA	Cotas sobre la superficie del agua	17	1	0	0
	Curvas de nivel	2	4	1	0
	Puntos de control horizontal	12	6	0	0
	Punto cota fija	16	0	0	0
	Punto más alto	15	5	0	0
INDUSTRIA	Area minera	1	0	5	2
	Límite de área industrial	60	13	2	3
	Cantera, mina	5	10	1	0
FISIOGRAFIA	Arena(seca)	30	4	5	2

	Zanja	3	12	1	0
	Lodo	4	70	1	0
	Cueva o caverna	20	9	0	0
AERONAUTIC 0	Límite aeroportuario(línea de contorno)	60	5	0	2
	Pista de longitud conocida	1	7	4	0
	Pista de longitud desconocida	4	5	0	0
	Pista sin pavimentar	9	1	4	0
	Helipuerto	7	5	0	0
LIMITES	Límite internacional	1	10	2	7
	Límite de reservas y parques	6	12	2	3
	Mojón limítrofe(hito)	10	11	0	0
COSTEROS	Malecón, muro de contención	27	5	1	0
	Muelle, embarcadero	26	0	1	0
	Bajo de anteplaya (arena)	3	9	1	1
	Arrecife	2	6	1	0
	Roca, arrecife sumergido	6	12	0	0
	Roca a flor de agua	9	8	0	0
	Naufragio al descubierto	7	7	2	1
	Plataforma de pozos petroleros	4	0	0	0
	Boya de petróleo o gas	15	5	0	0
CULTURAL	Estructura a escala	21	3	5	3
	Ruinas	31	9	0	5
	Antena de radio o mástil/menos de 61 m	35	5	1	0
	Estadio y similares(área prominente)	57	2	1	5

	Cementerio	58	4	0	5
	Límite de zona urbanizada	60	14	2	0
	Oleoducto visible	42	14	1	0
	Oleoducto subterráneo	43	14	1	6
	Línea de transmisión de energía	50	5	2	0
	Muro a lo largo del camino	52	9	1	5
	Muro de piedra(Muro)	53	13	1	0
	Palizada	54	12	1	4
	Alambrada (cerca)	55	6	1	7
	Polígono de tiro, ferias, campos de golf, etc.	38	6	1	7
	Represa, dique de mampostería	40	0	2	0
	Dique de tierra	12	10	1	0
	Casas ó edificaciones	5	0	0	0
	Iglesia	10	4	0	0
	Santuario	10	4	0	0
	Escuela	11	5	0	0
	Choza	36	4	0	0
	Hospital	22	5	0	2
	Edificio en ruinas	30	14	0	0
	Faro	32	5	0	0
	Monumento	33	5	0	0
	Estación de transformación de energía	34	5	0	0
	Tanque	46	5	0	0
	Tanque elevado	47	5	0	0
	Esclusa o compuerta	59	0	2	
	Línea de teléfono/telégrafo	51	5	1	

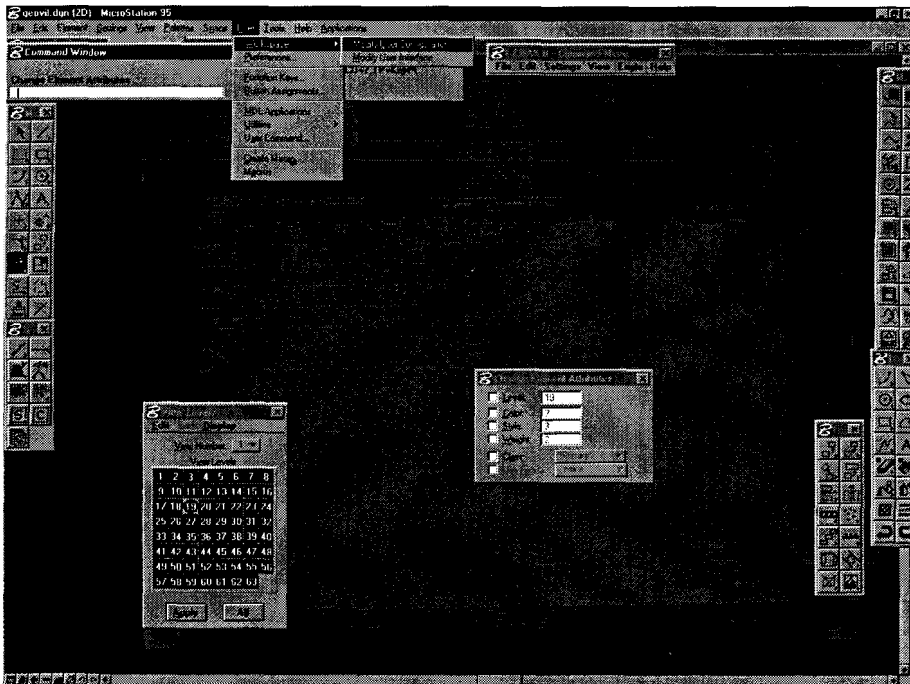
E.6 PASOS PARA DIGITALIZACION

Previo de haber realizado los pasos de georeferenciación procedemos a la digitalización de los archivos comprendidos para el proyecto.

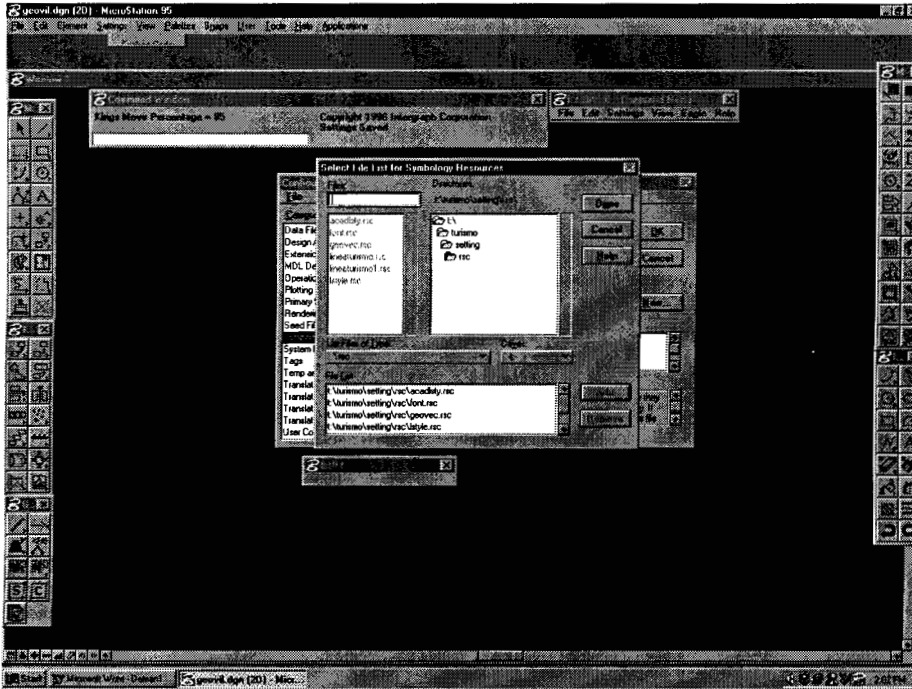
Verificar que se encuentren cargados los siguientes archivos, que corresponden a los features, celdas y tipos de líneas que han sido personalizados para el proyecto.



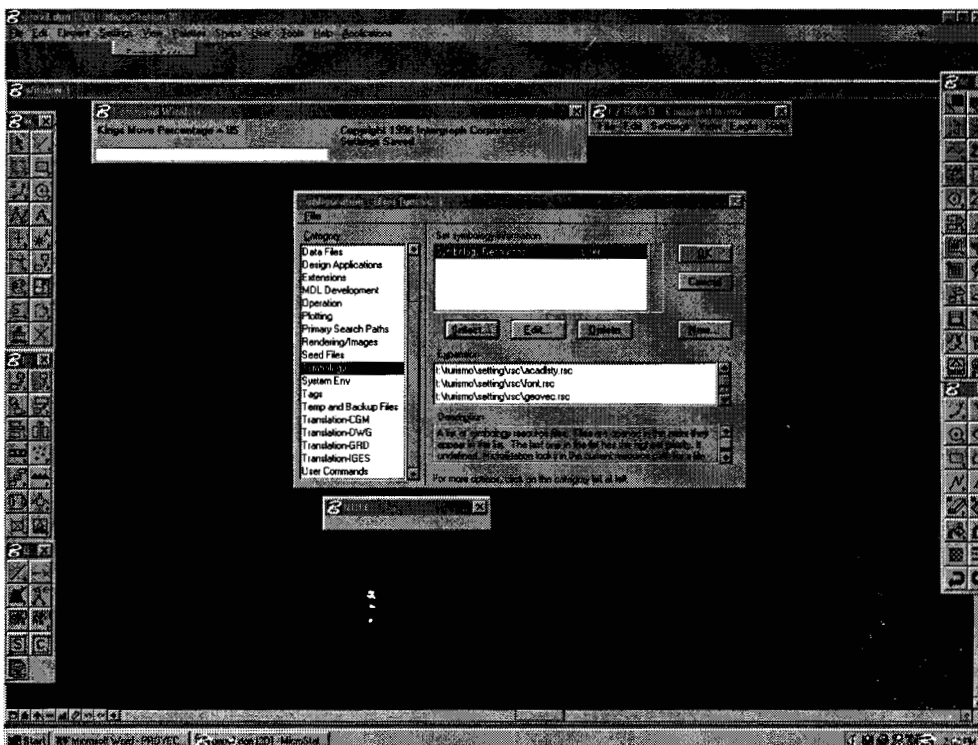
En el menú **User** escogemos la opción **Workspace** -> **Modify User Configuration** para cargar los tres archivos bases.



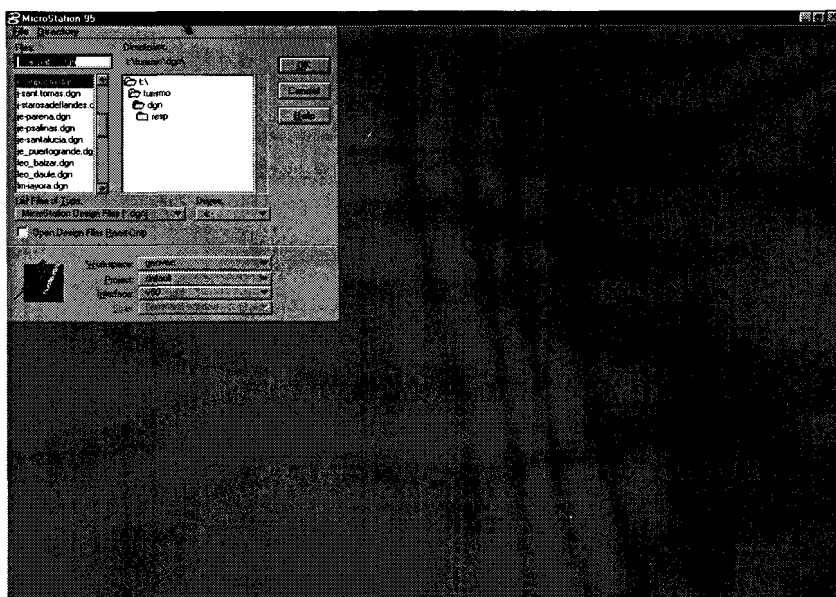
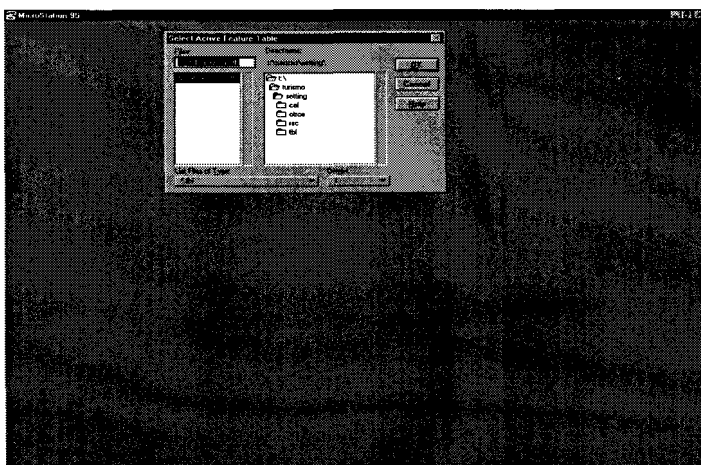
Luego, aparece la ventana de configuración donde debo elegir Simbology para seleccionar cada archivo añadiéndolos uno por uno.



Después de haber añadido cada archivo ejecutar el comando Done para que sean cargados en cada hoja de trabajo.

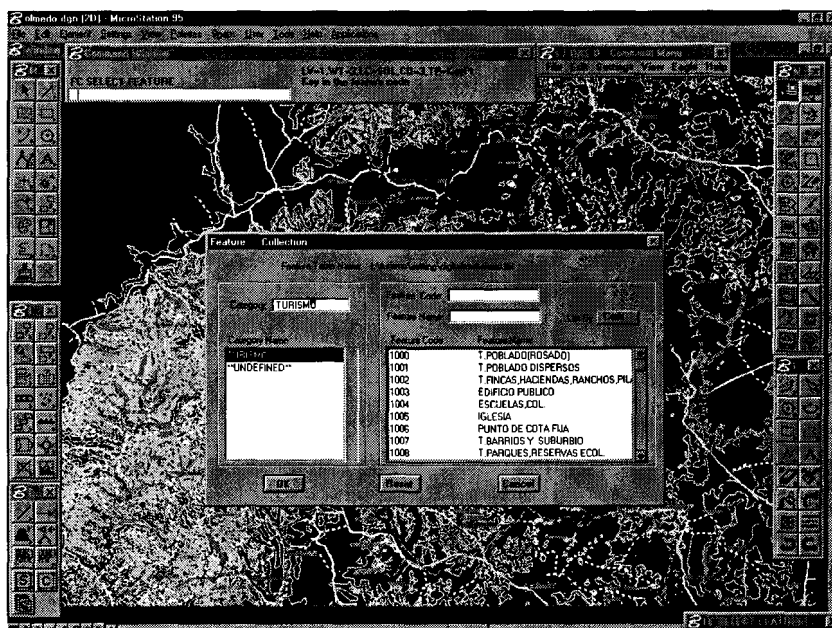


Los archivos mencionados anteriormente deben ser guardados en el directorio settings del servidor donde se encuentren creados los archivos de MicroStation, así cada vez que se cree un nuevo archivo o se trabaje sobre un archivo ya creado deben elegirse esos archivos al abrir y cargarlos en caso de ser nuevos siguiendo los pasos descritos en la parte superior.



Una vez cargados los archivos, para digitalizar fácilmente sin tener que setear a cada momento la capa activa sobre la cual vamos a trabajar podemos hacerlo automáticamente seteando los features que contienen las características de cada elemento a digitalizar y en que nivel se encuentra cada uno, así utilizamos las herramientas sean de MicroStation ó Igeovec eligiendo en Feature con que elemento vamos a trabajar.

E.6.1 DIGITALIZACION DE CURVAS DE NIVEL

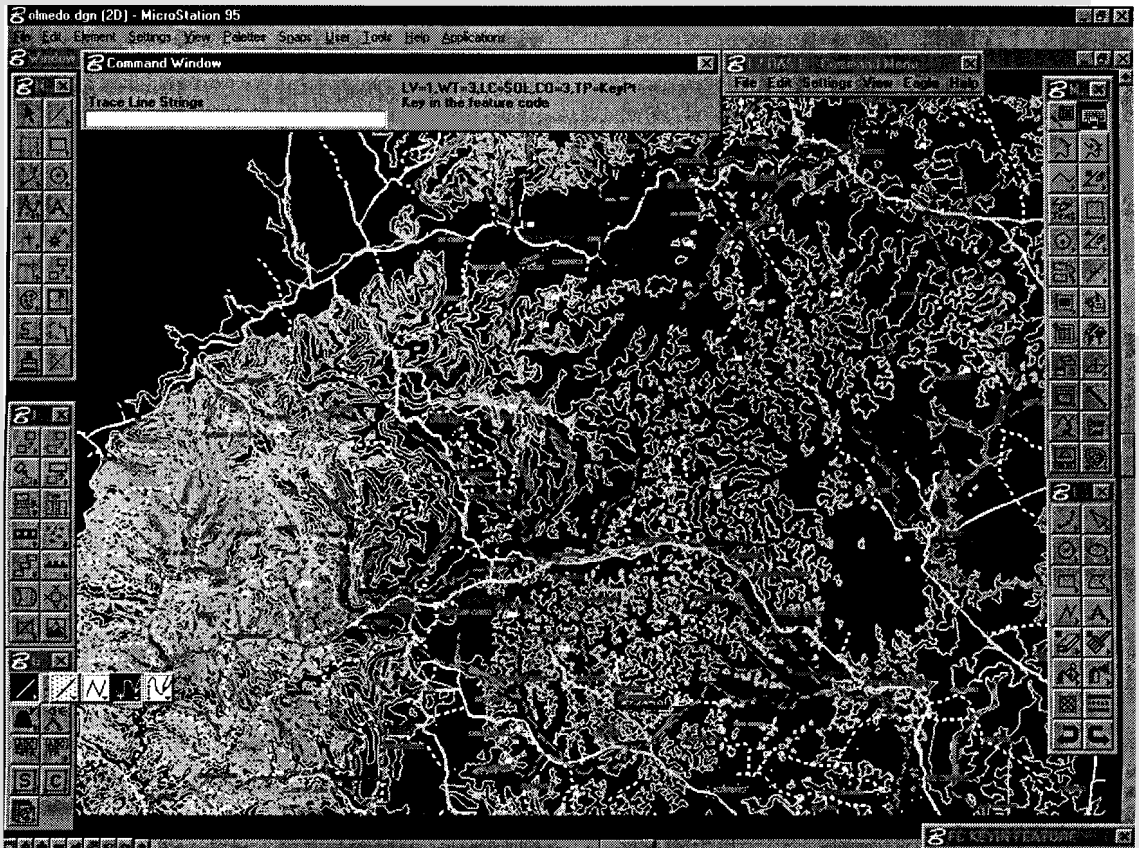


La digitalización de curvas de nivel, puede ser realizada de dos formas:

- A través de MicroStation, utilizando la herramienta de **Line String** e ir dibujando digitalizando sobre el archivo raster manualmente.

- b. Utilizando las herramientas de IGeovec, que están diseñadas específicamente para trabajar sobre archivos rasters, es decir, previamente a tener abierto el archivo .cit sobre el cual se va a trabajar, escoger en la paleta la herramienta **Geovec Lines** dentro de la cual se deben configurar los siguientes parámetros Intersection (Direction: Straight), Vector Snap (Mode: Snap to and Break), Smooth and Filter (Option: Filter).

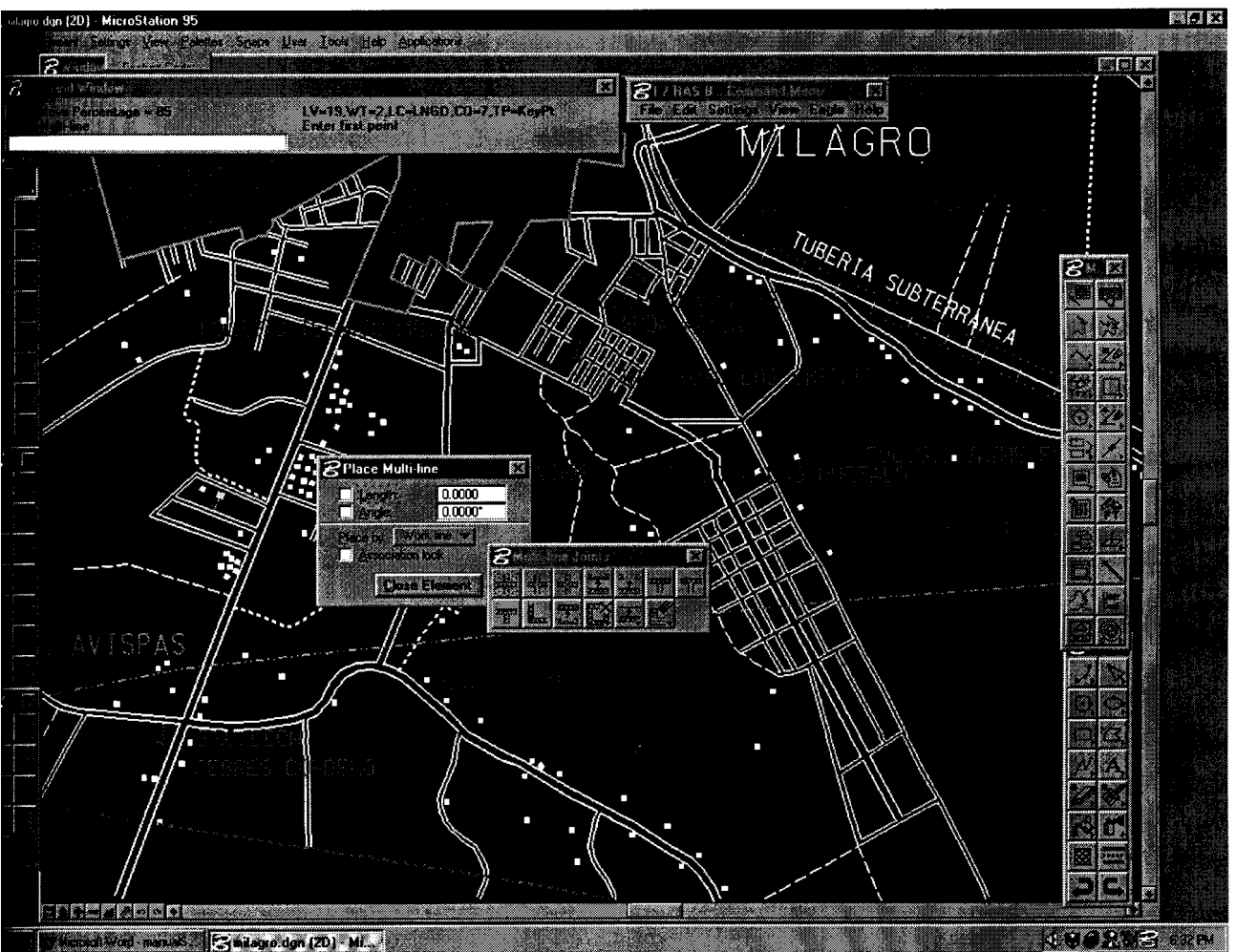




E.6.2 DIGITALIZACION DE VIAS

Para la digitalización de vías se ha utilizado una de las herramientas de MicroStation llamadas **MULTILINES**, las cuales han sido personalizadas para los propósitos del proyecto de acuerdo al tipo de línea visible sobre el mapa, se ha creado un archivo **.src** que contiene todos los tipos de líneas utilizados en el proyecto y que debe estar cargado previa digitalización.

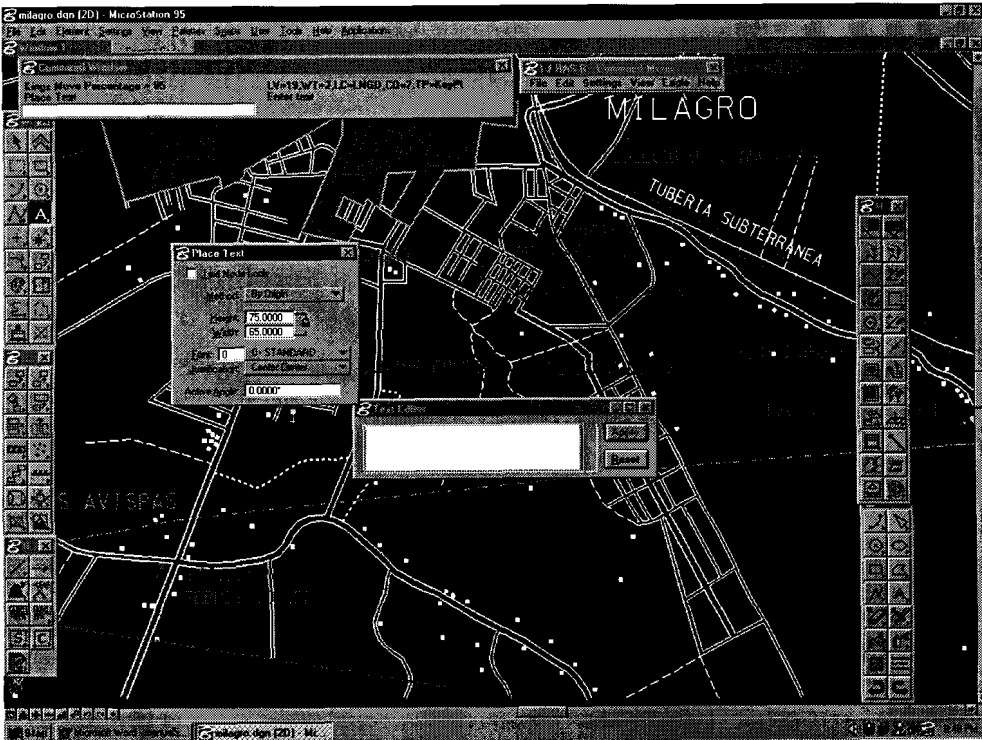
Dentro de la opción de Lines, escoger Multilines y arrastrar para sacar la paleta de opciones de multiline, por ejemplo la herramienta que sirve para cruce de carreteras y de acuerdo a cada situación en la que se encuentre el digitalizador deberá hacer uso de las múltiples opciones de éste tipo de línea tal como se muestra en la figura



E.6.3 COLOCACION DE TEXTOS

En el ítem se encuentra una tabla con todos los textos existentes dentro de cada nivel de información, para textos de nombre de ríos, escuelas, hoteles, barrios, etc. basta con utilizar la herramienta de MicroStation correspondiente a la colocación de textos

representada con una **A** seteando el método en **By Origin**, el **tamaño** del texto de acuerdo a cada capa, el tipo de **Font** a utilizar y la alineación.



Para lo que corresponde a la colocación de texto sobre curvas de nivel, en el área de Método se debe setear la opción **On Element** y seguir los mismos pasos mencionados anteriormente para los otros parámetros.

E.6.4 COLOCACION DE CELDAS

Se han determinado celdas para cada lugar turístico y establecimientos necesarios para brindar una mejor información.

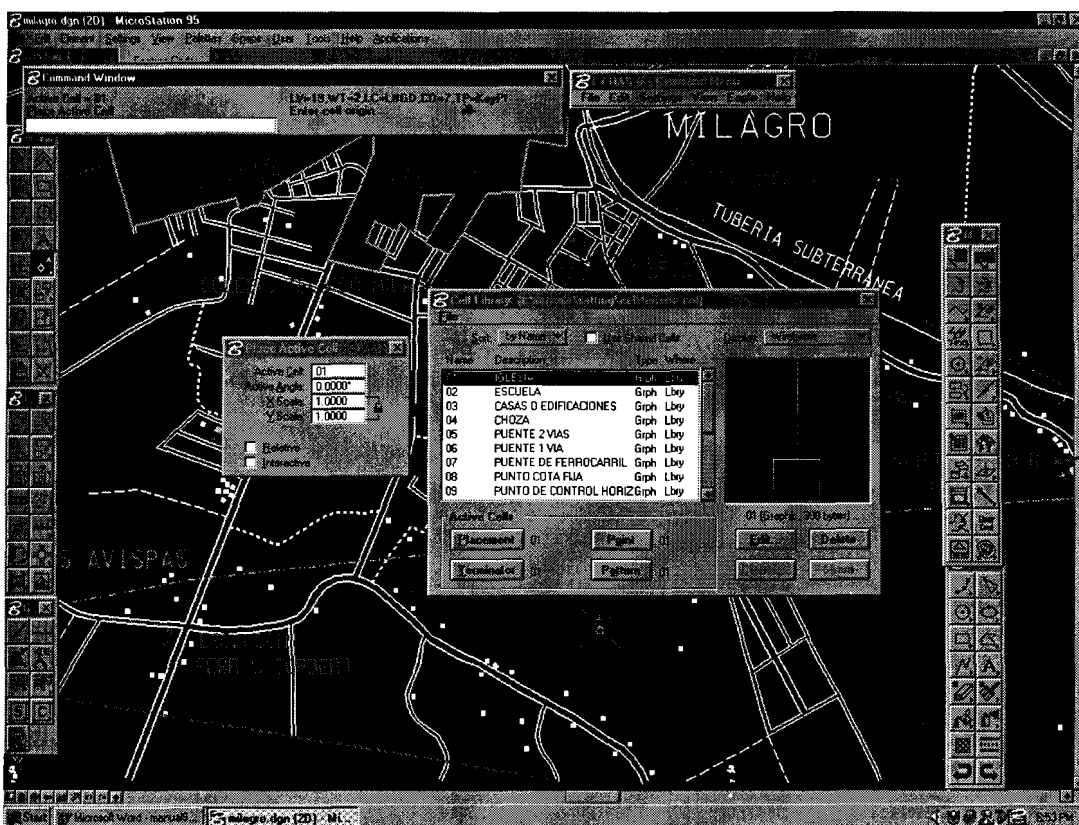
MicroStation nos ofrece la facilidad de poder crear celdas personalizadas y de la colocación de las mismas sobre los archivos .dgn

- Se procede primero a cargar el archivo:

Turismo.cel

- Luego, selecciona la celda que va a utilizar, accionando los botones de comando **Placement, Point, Terminator y Pattern**

Cargada la celda para la colocación en el .dgn escoger la herramienta **Place Active Cell** ubicándonos en la posición exacta.



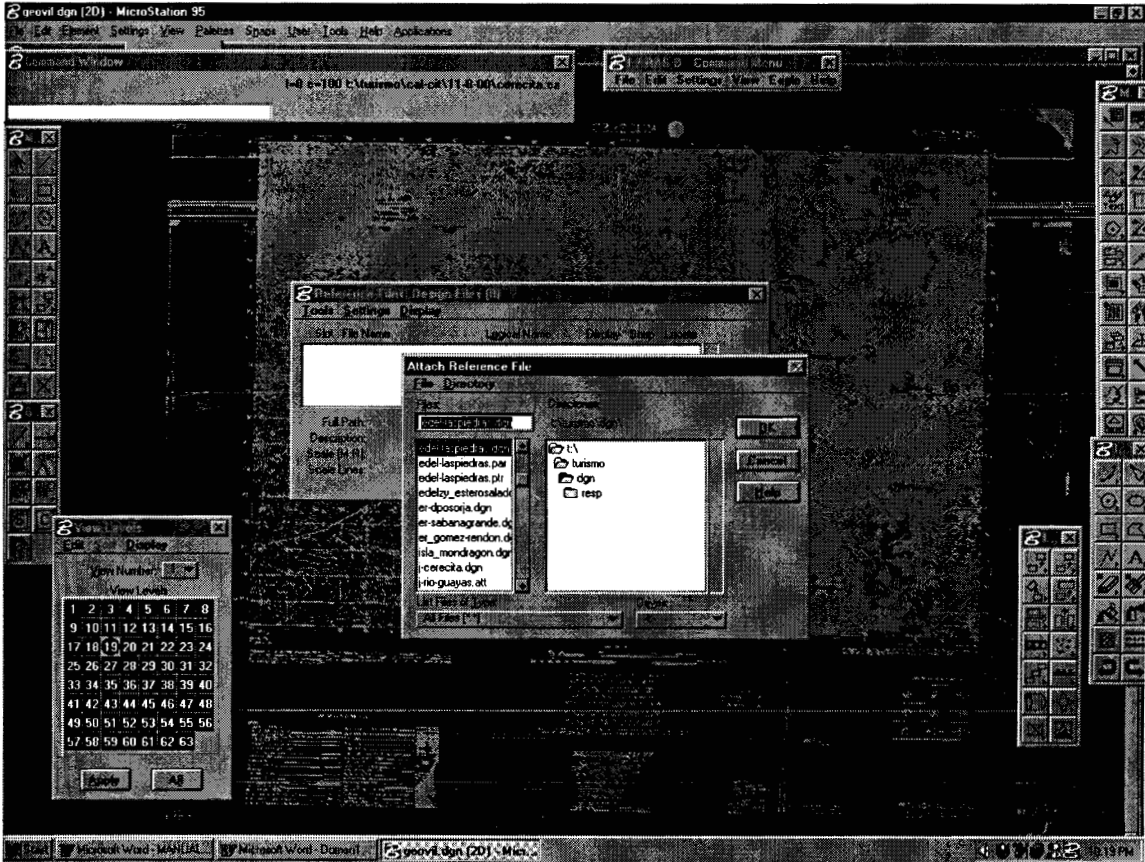


E.6.5 REFERENCIACION DE ARCHIVOS

Al terminar de digitalizar el archivo raster usando las herramientas antes mencionadas, se procede a referenciar; es decir verificar que la información que se digitalizó por separado coincidan entre si.

Se referencia todos los archivos que limitan con el archivo a revisar, verificando que tanto las curvas de nivel, carreteras, ríos, etc. posean continuidad entre ellos y representen toda la provincia del Guayas.

File - Reference - Tools - Attach -



E.7 LISTADO DE MAPAS COMPRENDIDOS EN DIGITALIZACION DE AREAS TURISTICAS DE LA PROVINCIA DEL GUAYAS

Número	Nombre
1	Guayas
2	Pichincha
3	Velasco Ibarra
4	Bocachico
5	Mocache
6	Olmedo
7	Balzar
8	Pedro Pablo Gómez
9	Colimes
	Palestina
10	Olon
12	Delicias
13	Guanábano
14	Pedro Carbo
15	Santa Lucía
16	Babahoyo
17	Manglaralto
	Las Piedras
18	

Número	Nombre
19	Isidro Ayora
20	Daule
21	Samborondon
	Simón Bolívar
22	San Pablo
24	Colonche
25	Balsas
26	Cerro Azul
27	Pascuales
28	Milagro
29	Naranjito
	Mathilde Esther
30	Salinas
32	Zapotal
33	Cerecita
34	Chongón
35	Guayaquil
36	Pedro J. Montero
	El Triunfo
37	Cumandá
39	Santo Tomás

Número	Nombre
40	Chanduy
	Gómez Rendón
42	Sábana Grande
43	Río Guayas
44	Laguna El Canclón
	Troncal Suscal
46	
	Punta de Piedras
47	
48	Playas
49	Estero Salado
50	Isla Mondragón
51	Santa Rosa de Flandés
52	Pancho Negro
53	Data de Posorja
54	Posorja
55	Canal de Jambelí
56	Naranjal
	San Felipe de Molleturo
57	
	Punta Brava
58	
59	Puerto Grande
60	Tenguel

Número	Nombre
61	El Carmen de Pijilí
	Punta Salinas
62	

ANEXO F

*Tablas de la base de datos
Eden*

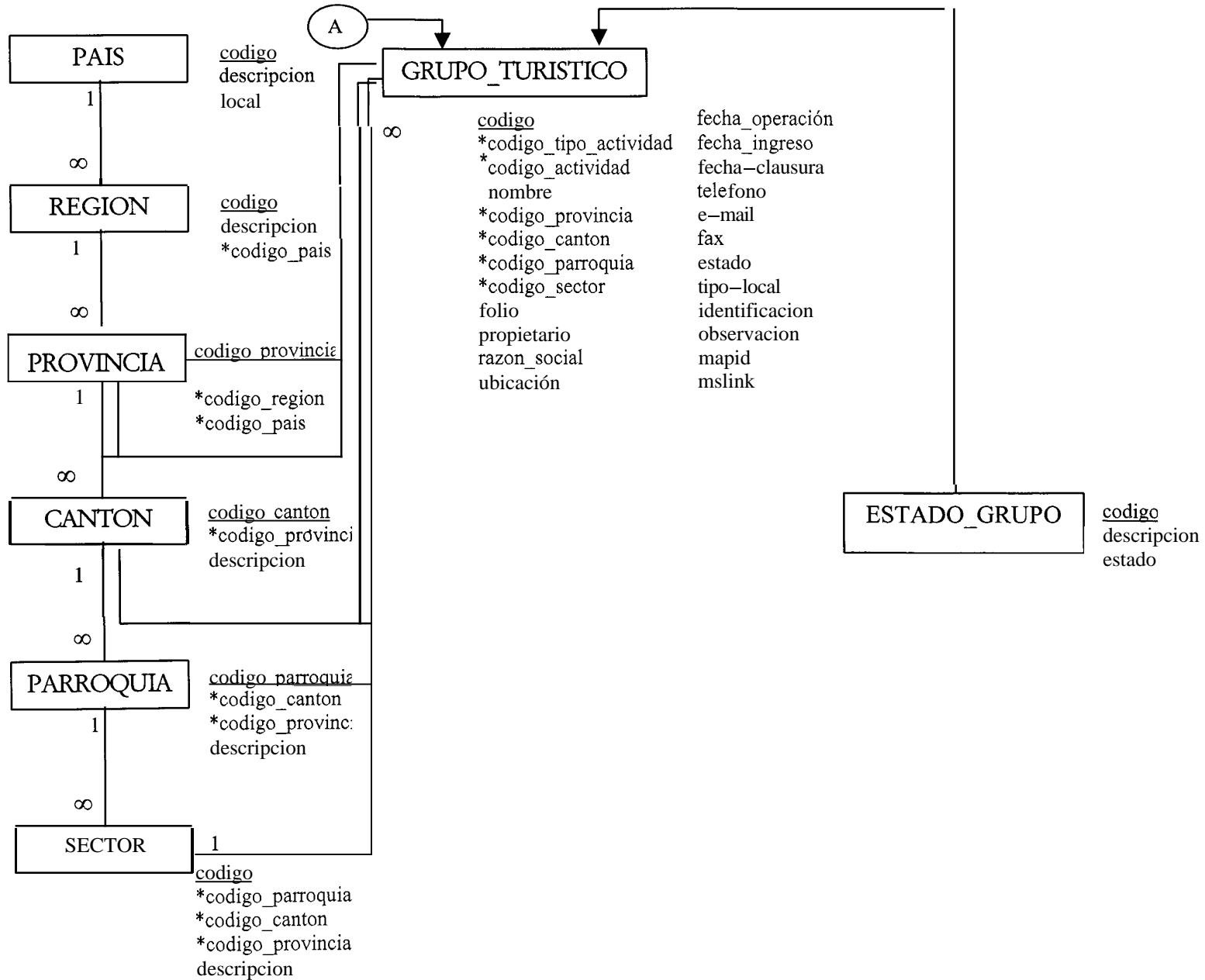
Tablas de la base de datos Eden

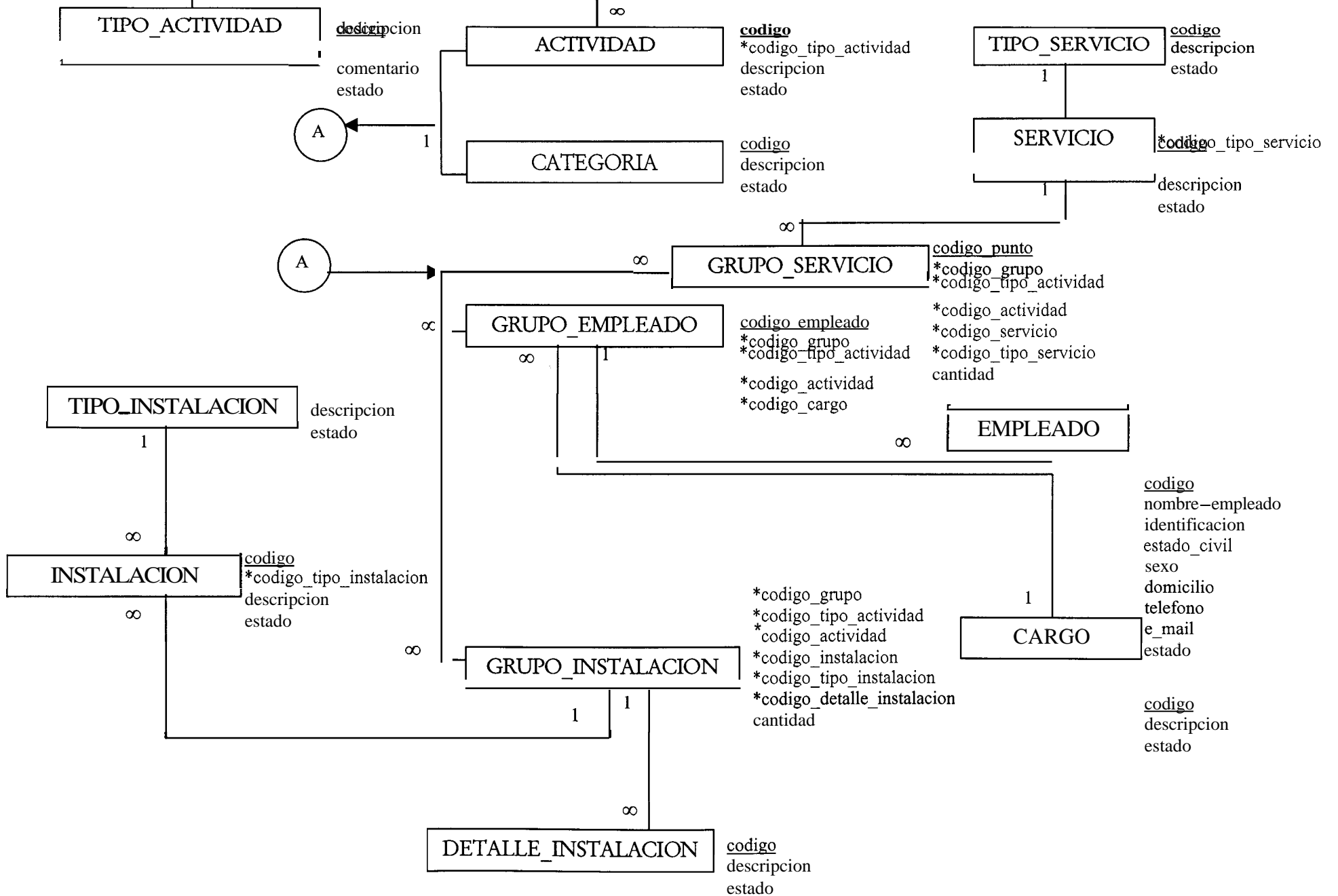
Tablas
actividad
canton
cargo
cartera
cartera-detalle
categoria
detalle-instalacion
empleado
estado-grupo
formaqago
grupo_empleado
grupo_instalacion
grupo-servicio
grupo-turistico
instalacion
motivoqago
pais
parroquia
provincia
región
sector
servicio
servicio_instalacion
tipo_actividad
tipo_instalacion
tipo_local
tipoqago
tipo_servicio
usuario

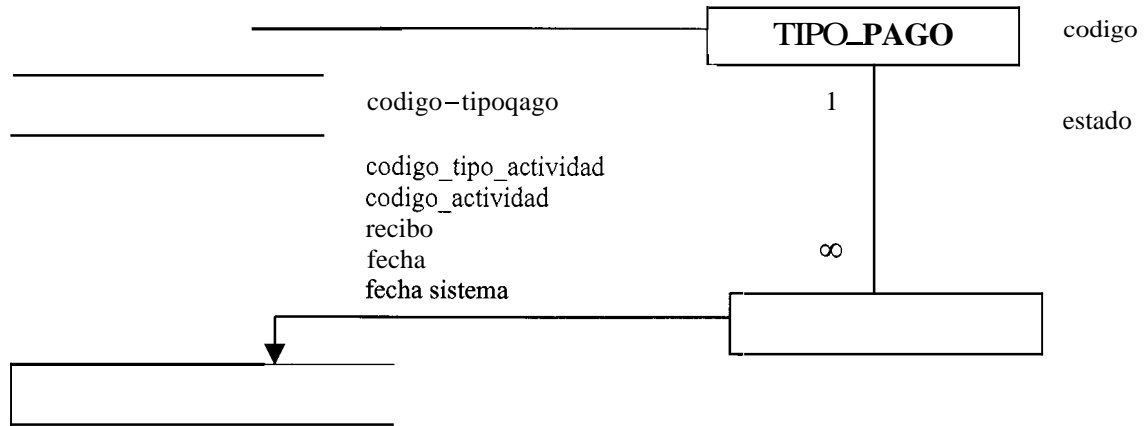
ANEXO G

Modelo Entidad/Relación
(E/R)

Diagrama de Entidad Relación







ANEXO H

Diccionario de Datos

H. Diccionario de Datos

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
cantidad	grupo-servicio	number	nn	cantidad del servicio	
cantidad	grupo-instalacion	number	nn	cantidad de la instalacion	
codigo	region	char[2]	pk	codigo alfanumérico	C, S, O e 1
codigo	sector	char[5]	pk	codigo alfanumérico	
codigo	pais	char[5]	pk	codigo alfanumérico	ECU, USA, etc.
codigo	cargo	char[3]	pk	codigo alfanumérico	
codigo	tipo-instalacion	char[5]	pk	codigo alfanumérico	
codigo	instalacion	char[5]	pk	codigo alfanumérico	
codigo	detalle_instalacion	char[10]	pk	codigo alfanumérico	
codigo	empleado	char[5]	pk	codigo alfanumérico	
codigo	motivoqago	char[5]	pk	codigo alfanumérico	
codigo	tipo_actividad	char[5]	pk	codigo alfanumérico	00 1,002, etc.
codigo	actividad	char[5]	pk	codigo alfanumérico	00 1,002, etc.
codigo	cartera	char[5]	pk	codigo alfanumérico	001, 002, etc.
codigo	cartera_detalle	char[5]	pk	codigo alfanumérico	001, 002, etc.
codigo	tipo_pago	char[5]	pk	codigo alfanumérico	
codigo	formaqago	char[5]	pk	codigo alfanumérico	

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
codigo		char[5]	pk	codigo alfanumérico	001,002, etc.
codigo		char[5]	pk	codigo alfanumérico	001,002, etc.
codigo	grupo_turistico	char[15]	pk	codigo alfanumérico	
codigo	tipo servicio	char[5]	pk	codigo alfanumérico	
codigo		char[10]	pk	codigo alfanumérico	
codigo	pagodet	char[10]	pk	codigo alfanumérico	
codigo_actividad	punto-servicio	char[5]	fk	codigo alfanumérico	
codigo_actividad	puntoempleado	char[5]	fk	codigo alfanumérico	
codigo_actividad	grupo-turistico	char[5]	pk	codigo alfanumérico	
codigo-actividad	punto-instalacion	char[5]	fk	codigo alfanumérico	
codigoactividad	pago	char[5]	fk	codigo alfanumérico	
codigoactividad	punto-turistico	char[5]	pk	codigo alfanumérico	
codigocanton	parroquia	char[3]	pk	codigo alfanumérico	
codigo-canton	sector	char[3]	pk	codigo alfanumérico	
codigocanton	grupo-turistico	char[3]	fk		
codigocanton	punto_turistico	char[3]	fk		
codigo_cargo	puntoempleado	char[3]	fk	codigo alfanumérico	
codigo_detalle_instalacio	punto_instalacion	char[10]	fk	codigo alfanumérico	

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
codigo_empleado	grupoempleado	char[5]	fk	codigo alfanumérico	
codigo_grupo	grupo_servicio	char[15]	fk	codigo alfanumérico	
codigo_grupo	grupoempleado	char[15]	fk	codigo alfanumérico	
codigo_grupo	grupo_instalacion	char[15]	fk	codigo alfanumérico	
codigo_grupo	pago	char[15]	fk	codigo alfanumérico	
codigo_grupo	grupo_turistico	char[15]	pk	codigo alfanumérico	
codigo_instalacion	grupo_instalacion	char[5]	fk	codigo alfanumérico	
codigo_objeto	grupo_turistico	char[10]	nn	codigo mapeado del mapa digitalizado	
codigoqais	region	char[3]	fk	codigo alfanumérico	ECU, USA, etc.
codigoqais	provincia	char[3]	fk	codigo alfanumérico	ECU, USA, etc.
codigoqarroquia	zona	char[3]	pk	codigo alfanumérico	
codigoqarroquia	grupo_turistico	char[3]	fk		
codigoqarroquia	punto_turistico	char[3]	fk		
codigoqropietario	grupo_turistico	char[50]	fk		
codigoqrovincia	canton	char[3]	pk	codigo alfanumérico	MNB, GUA, AZU, etc.
codigoqrovincia	parroquia	char[3]	pk	codigo alfanumérico	MNB, GUA, AZU, etc.

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
codigo_grovincia	zona	char[3]	pk	codigo alfanumérico	MNB, GUA, AZU, etc.
codigo_qrovincia	grupo_turistico	char[3]	fk		
codigo_provincia	grupo_turistico	char[3]	fk		
codigo_quinto	grupo_servicio	char[10]	fk	codigo alfanumérico	
codigo_quinto	grupo_empleado	char[10]	fk	codigo alfanumérico	
codigo_quinto	grupo_instalacion	char[10]	fk	codigo alfanumérico	
codigo_quinto	pago	char[10]	fk	codigo alfanumérico	
codigo_region	provincia	char[3]	fk	codigo alfanumérico	C, S, O e 1
codigo_representante	grupo_turistico	char[50]	fk		
codigo_servicio	grupo_servicio	char[5]	fk	codigo alfanumérico	
codigo_tipo_actividad	grupo_servicio	char[5]	fk	codigo alfanumérico	
codigo_tipo_actividad	punto_empleado	char[5]	fk	codigo alfanumérico	
codigo_tipo_actividad	actividad	char[5]	pk	codigo alfanumérico	001, 002, etc.
codigo_tipo_actividad	grupo_turistico	char[5]	pk	codigo alfanumérico	

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
codigo_tipo_actividad	grupo–instalacion	char[5]	fk	codigo alfanumérico	
codigo–tipo–actividad	pago	char[5]	fk	codigo alfanumérico	
codigo_tipo–actividad	grupo–turistico	char[5]	pk	codigo alfanumérico	
codigo_tipo–instalacion	instalacion	char[5]	pk	codigo alfanumérico	
codigo–tipo–instalacion	grupo–instalacion	char[5]	fk	codigo alfanumérico	
codigo_tipo–servicio	grupo–servicio	char[5]	fk	codigo alfanumérico	
codigo_tipo–servicio	servicio	char[5]	pk	codigo alfanumérico	001, 002, etc.
codigo_usuario	pagodet	char[10]	nn		
codigo_zona	grupo–turistico	char[3]	fk		
comentario	tipo__actividad	char[200]		comentario sobre la actividad	
descripcion	region	char[50]	nn	describe la región	Costa, Sierra, Oriente e Insular
descripcion	provincia	char[50]	nn	describe la provincia	Manabí, Guayas, Azuay, etc.
descripcion	canton	char[50]	nn	describe el canton	Guayaquil, Milagro, etc.
descripcion	parroquia	char[50]	nn	describe la parroquia	Ximena, Tarqui, etc.

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
descripcion	sector	char[50]	nn	describe la zona	Floresta, Esteros / Norte, Sur
descripcion	pais	char[50]	nn	describe el pais	Ecuador, Estados Unidos, etc.
descripcion	cargo	char[50]	nn	describe cargo	Cocinero, Administrador, etc.
descripcion	tipo_instalacion	char[50]	nn	describe la instalación	habitaciones, salas, auditorios, bancos etc.
descripcion	instalacion	char[50]	nn	describe la instalacion	sencillas, dobles, salas de espera, etc.
descripcion	detalle_instalacion	char[50]	nn	describe detalle de la instalacion	mesas, baños, aire acondicionado, butacas, etc.
descripcion	motivo_pago	char[50]	nn	describe motivo de pago	inversión, funcionamiento, registro, cambio propietario
descripcion	tipo_actividad	char[50]	nn	describe la actividad	Alojamiento, Agencias
descripcion	actividad	char[50]	nn	describe el tipo de actividad	Hotel, Hostería, etc.
descripcion	categoria	char[50]	nn	describe la categoría	Primera, Segunda, etc.
descripcion	servicio	char[50]	nn	describe servicio	Sauna, Lavandería, Vehículos, etc.
descripcion	tipo_servicio	char[50]	nn	describe tipo de servicio	básico, transporte, recreacional
descripcion	tipo_pago	number	nn	describe el tipo de pago	débito, crédito
descripción	forma_pago	char(50)	nn	describe la forma de pago	Cheque, efectivo, papeleta de deposito

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
domicilio	empleado	char[200]	nn		
e_mail	empleado	char[30]			
e_mail	grupo-turistico	char[30]			
estado	cargo	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	tipo_instalacion	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	instalacion	char[1]	nn	char[1]	
estado	detalle_instalacion	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	empleado	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	motivoqago	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	tipo_actividad	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	actividad	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	categoria	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	servicio	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado	grupo-turistico	char[1]	nn	estado del registro [activo, eliminado]	A, X

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
estado	formaqago	char(1)	Nn	estado del registro [activo, eliminado]	A, X
estado	tipo_servicio	char[1]	nn	estado del registro [activo, eliminado]	A, X
estado_civil	empleado	char[1]	nn	describe soltero, casado, etc.	S, C, D, V
fax	grupo-turistico	char[15]			
fecha	cartera	date	nn		
fecha_clausura	grupo-turistico	date	nn		
fecha-ingreso	grupo-turistico	date	nn		
fecha_operacion	grupo-turistico	date	nn		
folio	grupo-turistico	char[30]			
identificacion	empleado	char[15]	nn	cédula, pasaporte, etc.	
local	pais	char[1]	nn	codigo carácter	S, N
nombre	empleado	char[100]	nn	nombre del empleado	
nombre	grupo-turistico	char[50]	nn	describe grupo	McDonalds
password	usuario	Char(10)	nn	Password del usuario para ingresar a Eden	

Nombre del Campo	Nombre de la Tabla	Tipo de Dato	Atributo	Descripción	Valores
razon_social	grupo-turistico	char[50]	nn	EcuRestaurantes	
sexo	empleado	char[1]	nn	describe hombre, mujer	H, M
secuencial	usuario	number	nn		
telefono	empleado	char[15]			
telefono	grupo-turistico	char[15]			
ubicación	grupo-turistico	char[200]			
usuario	usuario	char(10)			
valor	Cartera_detalle	number	nn		

ANEXO I

Código Fuente

1.1 CODIGO FUENTE

MODULO GENERAL

```
Global vquery As String
Global vcodigo As String
Global vdescripcion As String
Global ruta_base As String
Global rutajpg As String
Global rutarpt As String
Global smotivo As String
Global user As String
Global password As String
Global vopcion As Integer
Sub main()
Call OpenTextFileTest
frmSplash.Show
frmLogin.Show vbModal
If frmLogin.LoginSucceeded = True Then
    user = frmLogin.txtUserName
    password = frmLogin.txtPassword
    Unload frmLogin
    MDIEden.Show
Else
    End
End If
End Sub
Sub llenar_servicio_instalacion()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
'Empieza transacción
db.BeginTrans
'actualizo
vquery = "delete from servicio-instalacion"
'Manda a ejecutar el query
db.Execute vquery
vquery = "select gs.codigo_grupo,gs.codigo_tipo_actividad,gs.codigo_actividad,"
vquery = vquery & " 'SERVICIOS' as tipo,ts.descripcion as tipo_servicio,s.descripcion as
servicio,null as detalle ,gs.cantidad"
vquery = vquery & " from grupo-servicio gs, tipo-servicio ts, servicio s"
vquery = vquery & " where gs.codigo_servicio=s.codigo and"
vquery = vquery & " gs.codigo_tipo_servicio=s.codigo_tipo_servicio and"
vquery = vquery & " gs.codigo_tipo_servicio = ts.codigo"
vquery = vquery & " UNION select
gi.codigo_grupo,gi.codigo_tipo_actividad,gi.codigo_actividad,"
vquery = vquery & " 'INSTALACIONES' as tipo,ti.descripcion as tipo_instalacion,i.descripcion
as instalacion,di.descripcion as detalle,"
vquery = vquery & " gi.cantidad"
vquery = vquery & " from grupo-instalacion gi, tipo-instalacion ti, instalacion i,
detalle-instalacion di"
vquery = vquery & " where gi.codigo_instalacion=i.codigo and"
vquery = vquery & " gi.codigo_tipo_instalacion=i.codigo_tipo_instalacion and"
vquery = vquery & " gi.codigo_tipo_instalacion=ti.codigo and"
vquery = vquery & " gi.codigo_detalle_instalacion=di.codigo;"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
```



```

While Not rs.EOF
    vquery = "insert into servicio-instalacion VALUES( "
    vquery = vquery & "" & rs.fields(0) & ""
    vquery = vquery & "," & rs.fields(1) & ""
    vquery = vquery & "," & rs.fields(2) & ""
    vquery = vquery & "," & rs.fields(3) & ""
    vquery = vquery & "," & rs.fields(4) & ""
    vquery = vquery & "," & rs.fields(5) & ""
    vquery = vquery & "," & rs.fields(6) & ""
    If Not IsNull(rs.fields(7)) Then
        vquery = vquery & "," & rs.fields(7) & ")"
    Else
        vquery = vquery & "," & 0 & ")"
    End If
    db.Execute vquery
    rs.MoveNext
Wend
rs.Close
'Manda a ejecutar el query
'db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Sub
Sub OpenTextFileTest()
Dim Micadena, Mitipo, Miruta, Mititulo
Open App.Path & "\eden.ini" For Input As #1 ' Abre el archivo ini para recibir los datos de rutas y
archivos.
rutajpg = Empty
Do While Not EOF(1) ' Repite el bucle hasta el final del archivo.
    Input #1, MiCadena ' Lee el carácter en dos variables.
    If InStr(1, Micadena, "[") <> 0 Then
Mititulo = Mid(MiCadena, InStr(1, Micadena, "[") + 1, InStr(1, Micadena, "]") - InStr(1,
Micadena, "[") - 1)
    Else
Select Case Mititulo
Case "conexión"
    Mitipo = Mid(MiCadena, 1, InStr(1, Micadena, "=") - 1)
    Miruta = Mid(MiCadena, InStr(1, Micadena, "=") + 1)
    Select Case Mitipo
    Case "path_base"
        ruta-base = Miruta
    End Select
Case "gráficos"
    Mitipo = Mid(MiCadena, 1, InStr(1, Micadena, "=") - 1)
    Miruta = Mid(MiCadena, InStr(1, Micadena, "=") + 1)
    Select Case Mitipo
    Case "path_jpg"
        rutajpg = Miruta
    End Select
    Case "reportes"
    Mitipo = Mid(MiCadena, 1, InStr(1, Micadena, "=") - 1)
    Miruta = Mid(MiCadena, InStr(1, Micadena, "=") + 1)
    Select Case Mitipo
    Case "path_rpt"

```

```

        rutarpt = Miruta
    End Select
End Select
End If
'Input #1, MiCadena ' Lee el carácter en dos variables.
Loop
Close #1 ' Cierra el archivo.
End Sub
Public Function duplicado() As Boolean
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.fields(0) = 0 Then
    duplicado = False
Else
    duplicado = True
End If
rs.Close
db.Close
End Function
Public Sub llenacombo(ByRef combo As ComboBox, icodigos As Integer)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
While Not rs.EOF
    combo.AddItem rs.fields(0) & " " & rs.fields(1)
    rs.MoveNext
Wend
rs.Close
db.Close
End Sub
Public Function cargacombo(numcod As Integer) As String
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim elementos As String
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
elementos = Chr$(9)
db.CursorLocation = adUseClient
db.Open ruta_base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
While Not rs.EOF
    If numcod = 1 Then
        elementos = elementos + rs.fields(0) + " " + rs.fields(1) + Chr$(9)
    Else
        If numcod = 2 Then
            elementos = elementos + rs.fields(0) + "-" + rs.fields(1) + " " + rs.fields(2) + Chr$(9)
        End If
    End If
    rs.MoveNext
Wend
rs.Close

```

```

db.Close
cargacombo= elementos
End Function
Sub llenar_grupo_catastro()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim rsi As ADODB.Recordset
Dim dbi As ADODB.Connection
Dim v_hb As Integer
Dim v_hbpl As Integer
Dim v_me As Integer
Dim v_mepl As Integer
Dim v_pe As Integer
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
Set rsi = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
'Empieza transacción
db.BeginTrans
'borro la tabla
vquery = "delete from grupo-turistico-catastro"
'Manda a ejecutar el query
db.Execute vquery
vquery = "select g.codigo,g.codigo_tipo_actividad,g.codigo_actividad,"
vquery = vquery & "t.descripcion,a.descripcion,g.representante,g.nombre,"
vquery = vquery & "p.descripcion,c.descripcion,pa.descripcion,g.ubicacion,"
vquery = vquery & "g.telefono,g.folio,ca.descripcion,g.identificacion,g.valor "
vquery = vquery & "from grupo-turistico,t,actividad a,"
vquery = vquery & "provinciap,canton c,parroquia pa,categoria ca "
vquery = vquery & "where g.codigo_tipo_actividad=t.codigo and "
vquery = vquery & "g.codigo_tipo_actividad=a.codigo_tipo_actividad and "
vquery = vquery & "g.codigo_actividad=a.codigo and "
vquery = vquery & "g.codigo_provincia=p.codigo_provincia and "
vquery = vquery & "g.codigo_provincia=c.codigo_provincia and "
vquery = vquery & "g.codigo_canton=c.codigo_canton and "
vquery = vquery & "g.codigo_provincia=pa.codigo_provincia and "
vquery = vquery & "g.codigo_canton=pa.codigo_canton and "
vquery = vquery & "g.codigo_parroquia=pa.codigo_parroquia and "
vquery = vquery & "g.codigo_categoria=ca.codigo and "
vquery = vquery & "g.estado='A'"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
While Not rs.EOF
vquery = "insert into grupo-turistico-catastroVALUES( "
vquery = vquery & "" & rs.fields(0) & ""
vquery = vquery & "," & rs.fields(1) & ""
vquery = vquery & "," & rs.fields(2) & ""
vquery = vquery & "," & rs.fields(3) & ""
vquery = vquery & "," & rs.fields(4) & ""
vquery = vquery & "," & rs.fields(5) & ""
vquery = vquery & "," & rs.fields(6) & ""
vquery = vquery & "," & rs.fields(7) & ""
vquery = vquery & "," & rs.fields(8) & ""
vquery = vquery & "," & rs.fields(9) & ""
vquery = vquery & "," & rs.fields(10) & ""
vquery = vquery & "," & rs.fields(11) & ""
vquery = vquery & "," & rs.fields(12) & ""
vquery = vquery & "," & rs.fields(13) & ""
vquery = vquery & "," & v_hb & ""

```

```

vquery = vquery & "," & v_hbpl & ""
vquery = vquery & "," & v_me & ""
vquery = vquery & "," & v_mepl & ""
vquery = vquery & "," & v_pe & ""
vquery = vquery & "," & rs.fields(14) & ""
vquery = vquery & "," & rs.fields(15) & ");"
db.Execute vquery
rs.MoveNext
Wend
rs.Close
"Termina la transacción
db.CommitTrans
db.Close
'Empieza transacción para actualizar instalaciones
db.CursorLocation = adUseClient
db.Open ruta-base
db.BeginTrans
vquery = "select codigo,codigo_tipo_actividad,codigo_actividad "
vquery = vquery & "from grupo-turistico"
vquery = vquery & "where estado='A'"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
'dbi.BeginTrans
While Not rs.EOF
'inicializo variables
v_me = 0
v_mepl = 0
v_hb = 0
v_hbpl = 0
v_pe = 0
'obtengo datos de mesa
query = "select cantidad from grupo-instalacion"
query = query & "where codigo-grupo = " & rs.fields(0) & "" and "
query = query & "codigo-tipo-actividad = " & rs.fields(1) & "" and "
query = query & "codigo_actividad = " & rs.fields(2) & "" and "
query = query & "codigo_tipo_instalacion='40' and codigo_instalacion='40' and "
query = query & "codigo_detalle_instalacion='40';"
rsi.Open query, db, adOpenStatic, adLockOptimistic
If rsi.EOF = False Then
If Not IsNull(rsi.fields(0)) Then
v_me = rsi.fields(0)
Else
v_me = 0
End If
End If
rsi.Close
'obtengo datos de plaza mesas
query = "select cantidad from grupo-instalacion"
query = query & "where codigo-grupo = " & rs.fields(0) & "" and "
query = query & "codigo-tipo-actividad = " & rs.fields(1) & "" and "
query = query & "codigo-actividad = " & rs.fields(2) & "" and "
query = query & "codigo_tipo_instalacion='41' and codigo_instalacion='41' and "
query = query & "codigo_detalle_instalacion='41';"
rsi.Open query, db, adOpenStatic, adLockOptimistic
If rsi.EOF = False Then
If Not IsNull(rsi.fields(0)) Then
v_mepl = rsi.fields(0)
Else
v_mepl = 0
End If
End If
End If

```

```

rsi.Close
'obtengo datos de habitaciones
query = "select cantidad from grupo-instalacion"
query = query & "where codigo-grupo = " & rs.fields(0) & " and "
query = query & "codigo-tipo-actividad= " & rs.fields(1) & " and "
query = query & "codigo-actividad= " & rs.fields(2) & " and "
query = query & "codigo_tipo_instalacion='42' and codigo_instalacion='42' and "
query = query & "codigo_detalle_instalacion='42';"
rsi.Open query, db, adOpenStatic, adLockOptimistic
If rsi.EOF = False Then
    If Not IsNull(rsi.fields(0)) Then
        v_hb = rsi.fields(0)
    Else
        v_hb = 0
    End If
End If
rsi.Close
'obtengo datos de habitaciones
query = "select cantidad from grupo-instalacion"
query = query & "where codigo-grupo = " & rs.fields(0) & " and "
query = query & "codigo-tipo-actividad= " & rs.fields(1) & " and "
query = query & "codigo-actividad= " & rs.fields(2) & " and "
query = query & "codigo_tipo_instalacion='43' and codigo_instalacion='43' and "
query = query & "codigo_detalle_instalacion='43';"
rsi.Open query, db, adOpenStatic, adLockOptimistic

If rsi.EOF = False Then
    If Not IsNull(rsi.fields(0)) Then
        v_hbpl = rsi.fields(0)
    Else
        v_hbpl = 0
    End If
End If
rsi.Close
'obtengo datos de plaza empleados
query = "select cantidad from grupo-instalacion"
query = query & "where codigo_grupo = " & rs.fields(0) & " and "
query = query & "codigo-tipo-actividad= " & rs.fields(1) & " and "
query = query & "codigo-actividad= " & rs.fields(2) & " and "
query = query & "codigo_tipo_instalacion='44' and codigo_instalacion='44' and "
query = query & "codigo_detalle_instalacion='44';"
rsi.Open query, db, adOpenStatic, adLockOptimistic
If rsi.EOF = False Then
    If Not IsNull(rsi.fields(0)) Then
        v_pe = rsi.fields(0)
    Else
        v_pe = 0
    End If
End If
rsi.Close
vquery = "update grupo-turistico-catastro"
vquery = vquery & "set hb=" & v_hb & ","
vquery = vquery & "hb_pl=" & v_hbpl & ","
vquery = vquery & "me=" & v_me & ","
vquery = vquery & "me_pl=" & v_mepl & ","
vquery = vquery & "pe=" & v_pe & " "
vquery = vquery & "where codigo = " & rs.fields(0) & " and "
vquery = vquery & "codigo-tipo-actividad= " & rs.fields(1) & " and "
vquery = vquery & "codigo-actividad= " & rs.fields(2) & " ";"
db.Execute vquery

```



```

    rs.MoveNext
Wend
'dbi.CommitTrans
'rsi.Close
rs.Close
'Termina la transacción
db.CommitTrans
db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub
Sub calculo_valores()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
Set rsi = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
'Empieza transacción para actualizar valores, según la actividad, categoría
db.BeginTrans
    vquery = "select codigo_tipo_actividad,codigo_actividad,codigo_categoria,valor "
    vquery = vquery & "from valor-actividad-categoria"
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    While Not rs.EOF
        vquery = "update grupo-turistico "
        If Not IsNull(rs.fields(3)) Then
            vquery = vquery & "set valor=" & rs.fields(3) & " "
        Else
            vquery = vquery & "set valor-" & 0 & " "
        End If
        vquery = vquery & "where codigo_tipo_actividad = " & rs.fields(0) & " and "
        vquery = vquery & "codigo_actividad = " & rs.fields(1) & " and "
        vquery = vquery & "codigo_categoria = " & rs.fields(2) & " ;"
        db.Execute vquery
        rs.MoveNext
    Wend
    rs.Close
'Termina la transacción
db.CommitTrans
db.Close
'Mensaje de finalización
    MsgBox "El cálculo de los valores finalizó correctamente", vbInformation, "Sistema EDEN"
Exit Sub
AddErr:
    MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub

```

MDIEden Codigo

```
Private Sub ayuace_Click()
    frmAcercade.Show
End Sub
Private Sub infgeogua_Click()
    vopcion = 0
    frm_mapa.Show
End Sub
Private Sub infimpact_Click()
    CryReportMdi.ReportFileName = rutarpt & "\tipo_actividad.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimpcat_Click()
    CryReportMdi.ReportFileName = rutarpt & "\categoria.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimpcat1_Click()
    CryReportMdi.ReportFileName = rutarpt & "\grupo_turistico_catastro.rpt"
    'tabla temporal
    Cali llenar_grupo_catastro
    CryReportMdi.Action = True
End Sub
Private Sub infimpgrudet_Click()
    CryReportMdi.ReportFileName = rutarpt & "\grupo_turistico_anexos.rpt"
    'tabla temporal
    Call llenar_servicio_instalacion
    CryReportMdi.Action = True
End Sub
Private Sub infimpgrugru_Click()
    CryReportMdi.ReportFileName = rutarpt & "\grupo_turistico.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimpgrupar_Click()
    Frmreportes.Show
End Sub
Private Sub infimpins_Click()
    CryReportMdi.ReportFileName = rutarpt & "\tipo_instalacion.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimppago_Click()
    CryReportMdi.ReportFileName = rutarpt & "\grupo_turistico_pagos.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimppro_Click()
    CryReportMdi.ReportFileName = rutarpt & "\provincias.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infimpser_Click()
    CryReportMdi.ReportFileName = rutarpt & "\tipo_servicio.rpt"
    CryReportMdi.Action = True
End Sub
Private Sub infturgru_Click()
    frmgrupo.Show
End Sub
Private Sub infturreg_Click()
    frmgrupopago.Show
End Sub
Private Sub inftursal_Click()
    ' termina la ejecución de la aplicación
```

```
End Sub
Private Sub manact_Click()
frmactividad.Show
End Sub
Private Sub mancan_Click()
frmcanton.Show
End Sub
Private Sub mancat_Click()
frmcategoria.Show
End Sub
Private Sub manconcal_Click()
Call calculo_valores
End Sub
Private Sub mandetins_Click()
frmdetalleinst.Show
End Sub
Private Sub manempleados_Click()
frmempleado.Show
End Sub
Private Sub manestados_Click()
frmestadogrupo.Show
End Sub
Private Sub manins_Click()
frminstalacion.Show
End Sub
Private Sub manmotivopago_Click()
frmmotivopago.Show
End Sub
Private Sub manpar_Click()
frmparroquial.Show
End Sub
Private Sub manpro_Click()
frmprovincia.Show
End Sub
Private Sub mansec_Click()
frmsector.Show
End Sub
Private Sub manser_Click()
frmservicio.Show
End Sub
Private Sub mantinfcontforma_Click()
frmformapago.Show
End Sub
Private Sub mantipoact_Click()
frmtipoactividad.Show
End Sub
Private Sub mantipoins_Click()
frmtipoinstalacion.Show
End Sub
Private Sub mantipopago_Click()
frmtipopago.Show
End Sub
Private Sub mantiposer_Click()
frmtiposervicio.Show
End Sub
Private Sub mnucambioclave_Click()
frmLogin1.Show
End Sub
```

frm_mapa

Option Explicit

Private WithEvents PrimaryCLS As clsconton

Dim mbChangedByCode As Boolean

Dim mvBookMark As Variant

Dim mbEditFlag As Boolean

Dim mbAddNewFlag As Boolean

Dim mbDataChanged As Boolean

Dim fotoruta As String

Dim xbuff1 As Integer

Dim ybuff1 As Integer

Dim xbuff2 As Integer

Dim ybuffs As Integer

Dim ZoomFlag As Integer

Private Sub cmdabrir_Click()

Dim sFileName As String

Me.cdlgabrir.FileName = ""

' Open the File Open dialog Box

Me.cdlgabrir.ShowOpen

sFileName = Me.cdlgabrir.FileName

'if file name is selected

If Len(sFileName) > 0 Then

 Caption = sFileName

 DrDwg1.DrawingName = sFileName

 DrDwg1.ReDraw

End If

End Sub

Private Sub DrDwg1_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)

' If mouse is down in the first time...

 If ZoomFlag = 1 Then

 ' Pickup the Mouse coordinates..

 xbuff1 = x / Screen.TwipsPerPixelX

 ybuff1 = y / Screen.TwipsPerPixelY

 ZoomFlag = 2

 End If

End Sub

Private Sub DrDwg1_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)

 Dim retcode As Integer

 Dim nVportNo As Integer

 ' ...Second time ...

 If ZoomFlag = 2 Then

 xbuff2 = x / Screen.TwipsPerPixelX

 ybuff2 = y / Screen.TwipsPerPixelY

 nVportNo = DrDwg1.GetActiveVPort()

 retcode = DrDwg1.ZoomWindow(nVportNo, xbuff1, ybuff1, xbuff2, ybuffs)

 If retcode = -212 Then

 MsgBox ("Invalid ViewPort specified or ViewPort number exceeds the bounds.")

 ElseIf retcode = -123 Then

 MsgBox ("Invalid drawing file.")

 End If

 ZoomFlag = 0

 End If

End Sub

Private Sub Form_Load()

 Dim rs As ADODB.Recordset

 Dim db As ADODB.Connection

 Dim sFileName As String

 'si entro por la pantalla de grupo

 If (opcion = 1) Then

```

Set db =New ADODB.Connection
Set rs =New ADODB.Recordset
db.CursorLocation =adUseClient
db.Open ruta-base
vquery = "select cstr(mapid) from grupo-turistico "
vquery = vquery & "where código=" & frmgrupo.Txtfields(0).Text & " and "
vquery = vquery & "codigo_tipo_actividad=" & Mid(frmgrupo.Cmbfields(1).Text, 1, InStr(1,
frmgrupo.Cmbfields(1).Text, " ") - 1) & " and "
vquery = vquery & "codigo_actividad=" & Mid(frmgrupo.Cmbfields(2).Text, 1, InStr(1,
frmgrupo.Cmbfields(2).Text, " ") - 1) & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
sFileName =rutajpg & "\" & rs.fields(0) & ".dwg"
'if file name is selected
If Len(sFileName) > 0 Then
Caption = sFileName
DrDwg1.DrawingName = sFileName
DrDwg1.ReDraw
End If
End If
End Sub
Private Sub Form_Resize()
Me.DrDwg1.Left = ScaleLeft
Me.DrDwg1.Top = ScaleTop
Me.DrDwg1.Width = ScaleWidth - 1.5 * Me.cmdabrir.Width
Me.DrDwg1.Height = ScaleHeight
Me.cmdabrir.Left = ScaleWidth - 1.2 * Me.cmdabrir.Width
End Sub

```

Frmactividad

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(2).Width = DataGrid1.Width * 0.5
DataGrid1.Columns(3).Width = DataGrid1.Width * 0.1
DataGrid1.Columns(0).Caption = "COD TIPO"
DataGrid1.Columns(1).Caption = "CODIGO"
DataGrid1.Columns(2).Caption = "DESCRIPCION"
DataGrid1.Columns(3).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i=0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
End Sub

Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i=0

```

```

For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocanton.ConnectionString = ruta-base
If Not oText.Text = "" Then
Select Case i
Case 0
vquery = "select descripcion from tipoactividad where codigo=" & Trim(oText.Text) & ""
Adocanton.RecordSource = vquery
Adocanton.Refresh
txtDescripcion(i).Text = Adocanton.Recordset.fields(0)
End Select
End If
i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
vquery = "select codigo,descripcion from tipoactividad ORDER BY DESCRIPCION"
frmbuscar.Col = 0
frmbuscar.Show vbModal
Me.txtFields(0) = vcodigo
Me.txtDescripcion(0) = vdescripcion
End Sub
Private Sub DataGrid1_Click()
End Sub
Private Sub Form_Load()
vquery = "select codigo_tipo_actividad,codigo,descripcion,estado from actividad Order by
descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub
Select Case KeyCode
Case vbKeyEscape
cmdClose_Click
Case vbKeyEnd
cmdLast_Click
Case vbKeyHome
cmdFirst_Click
Case vbKeyUp, vbKeyPageUp
If Shift = vbCtrlMask Then
cmdFirst_Click
Else
cmdPrevious_Click
End If
Case vbKeyDown, vbKeyPageDown
If Shift = vbCtrlMask Then

```

```

    cmdLast_Click
Else
    cmdNext_Click
End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmdbuscarprov.Visible = True
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
    Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    cmdbuscarprov.Visible = True
    mbEditFlag = True
    Call bloquear(True)
    SetButtons False
    Exit Sub

```

```

EditErr:
  MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
  On Error Resume Next
  mbAddNewFlag = False
  PrimaryCLS.Cancel
  PrimaryCLS.Requery
  Call FormatoGrid
  SetButtons True
  Call bloquear(False)
  cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
  On Error GoTo UpdateErr
  'HACER TODAS LAS VERIFICACIONES
  'Verificando que los campos no esten vacíos
  For Each oText In Me.txtFields
    If Trim(oText.Text) = Empty Then
      oText.SetFocus
      MsgBox "Campo vacío no se puede grabar", vbCritical
      Exit Sub
    End If
  Next
  vquery = "select count(*) from actividad where codigo_tipo_actividad=" & Trim(Me.txtFields(0))
  & """"
  vquery = vquery & " and código=" & Trim(Me.txtFields(1)) & """"
  If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
    Exit Sub
  End If
  'FIN DE VERIFICACIONES
  'graba registro nuevo
  If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
  End If
  i = 0
  For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
  Next
  mbAddNewFlag = False
  PrimaryCLS.Update
  Call bloquear(False)
  Me.cmdbuscarprov.Visible = False
  SetButtons True
  Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
  Unload Me
End Sub
Private Sub cmdFirst_Click()
  On Error GoTo GoFirstError
  PrimaryCLS.MoveFirst
  mbDataChanged = False
  Exit Sub

```




```

GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
    Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdRefresh.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 2 Then
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
        If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
            KeyAscii = 0
        End If
    End If
End If
End Sub

```

Frm_buscar

```

Private Sub Adobuscar_WillChangeRecordset(ByVal adReason As ADODB.EventReasonEnum,
adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
Me.DataGrid1.Col = Col
Me.Adobuscar.Caption = Me.DataGrid1.Text
End Sub
Private Sub cmdcancelar_Click()
vcodigo = Empty
Unload Me
End Sub
Private Sub DataGrid1_Db1Click()
Me.DataGrid1.Col = Col

```

```

vcodigo = Me.DataGrid1.Text
Col = Col + 1
Me.DataGrid1.Col = Col
vdescripcion = Me.DataGrid1.Text
Unload Me
End Sub
Private Sub Form_Load()
Adobuscar.ConnectionString = ruta-base
Adobuscar.RecordSource = vquery
Set DataGrid1.DataSource = Adobuscar
Adobuscar.Refresh
End Sub

```

frmcantón

```

Private WithEvents PrimaryCLS As clscantón
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(2).Width = DataGrid1.Width * 0.5
DataGrid1.Columns(0).Caption = "COD PROV"
DataGrid1.Columns(1).Caption = "CODIGO"
DataGrid1.Columns(2).Caption = "DESCRIPCION"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocantón.ConnectionString = ruta_base
If Not oText.Text = "" Then
Select Case i
Case 0
vquery = "select descripcion from provincia where codigo_provincia =" & Trim(oText.Text)
Adocantón.RecordSource = vquery
Adocantón.Refresh
txtDescripcion(i).Text = Adocantón.Recordset.fields(0)
End Select
End If
i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
frmbuscar.Col = 0

```

```

frmbuscar.Show vbModal
Me.txtFields(0) = vcodigo
Me.txtDescripcion(0) = vdescripcion
End Sub
Private Sub Form_Load()
    vquery = "select codigo_provincia,codigo_canton,descripcion from canton Order by descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
    cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro: " & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
    Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields

```

```

Exit Sub
End If
Next
vquery = "select count(*) from canton where codigo_provincia=" & Me.txtFields(0)
vquery = vquery & " and codigo_canton=" & Me.txtFields(1)
If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmdbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:

```

```

MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 2 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0
End If
End If
End If
End Sub

```

frmcategoria

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.5
DataGrid1.Columns(2).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(0).Caption = "COD"
DataGrid1.Columns(1).Caption = "DESCRIPCION"
DataGrid1.Columns(2).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocanton.ConnectionString = ruta_base
i = i + 1
Next
End Sub

```

```

Private Sub Form_Load()
    vquery = "select codigo,descripcion,estado from categoria Order by descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
    Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    Call bloquear(True)

```

```

lblStatus.Caption = "Agregar registro"
mbAddNewFlag = True
SetButtons False
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
PrimaryCLS.Delete
Call cargartexto
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr
lblStatus.Caption = "Modificar registro"
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
Call cargartexto
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from categoria where codigo='" & Me.txtFields(0) & "'"

```

```

If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
    Exit Sub
End If
FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
End If
i=0
For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
SetButtons True
Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
    Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
    Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal

```



```

cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 1 Then
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
        If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
            KeyAscii = 0
        End If
    End If
End If
End Sub

```

frmclausura

```

Private Sub cmdaceptar_Click(Index As Integer)
smotivo = Empty
If Trim(txtmotivo.Text) <> Empty Then
    smotivo = txtmotivo.Text
End If
Unload Me
End Sub

```

frmconsultar

```

Private Sub Adoconsultar_WillChangeRecordset(ByVal adReason As ADODB.EventReasonEnum,
adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
Me.DataGrid1.Col = Col
Me.Adoconsultar.Caption = Me.DataGrid1.Text
End Sub
Private Sub cmdcancelar_Click()
vcodigo = Empty
vquery = Empty
Unload Me
End Sub
Private Sub DataGrid1_Db1Click()
    Unload Me
End Sub
Private Sub Form_Load()
Adoconsultar.ConnectionString = ruta_base
Adoconsultar.RecordSource = vquery
Set DataGrid1.DataSource = Adoconsultar
Adoconsultar.Refresh
Col = 0
End Sub

```

frmdetalleinst

```

Private WithEvents PrimaryCLS As clscanton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String

```

```

Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.5
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0.22
    DataGrid1.Columns(0).Caption = "COD"
    DataGrid1.Columns(1).Caption = "DESCRIPCION"
    DataGrid1.Columns(2).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
    Dim i As Long
    i = 0
    For Each oText In Me.txtFields
        oText.Enabled = valor
        i = i + 1
    Next
End Sub
Private Sub cargartexto()
    Dim i As Long
    Dim a(1) As String
    i = 0
    For Each oText In Me.txtFields
        oText.Text = PrimaryCLS.fields(i)
        Adocanton.ConnectionString = ruta-base
        i = i + 1
    Next
End Sub
Private Sub Form_Load()
    vquery = "select codigo,descripcion,estado from detalle-instalacion Order by descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then

```

```

    cmdLast_Click
Else
    cmdNext_Click
End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
    Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    mbEditFlag = True
    Call bloquear(True)
    SetButtons False
    Exit Sub
EditErr:
    MsgBox Err.Description
End Sub

```

```

Private Sub cmdCancel_Click()
    On Error Resume Next
    mbAddNewFlag = False
    PrimaryCLS.Cancel
    PrimaryCLS.Requery
    Call FormatoGrid
    SetButtons True
    Call bloquear(False)
    Call cargartexto
    cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
    On Error GoTo UpdateErr
    'HACER TODAS LAS VERIFICACIONES
    'Verificando que los campos no esten vacíos
    For Each oText In Me.txtFields
        If Trim(oText.Text) = Empty Then
            oText.SetFocus
            MsgBox "Campo vacío no se puede grabar", vbCritical
            Exit Sub
        End If
    Next
    vquery = "select count(*) from detalle_instalacion where codigo=" & Me.txtFields(0) & """"
    If duplicado And mbAddNewFlag = True Then
        txtFields(0).SetFocus
        MsgBox "Registro duplicado", vbCritical
        Exit Sub
    End If
    'FIN DE VERIFICACIONES
    'graba registro nuevo
    If mbAddNewFlag = True Then
        PrimaryCLS.AddNew
    End If
    i = 0
    For Each oText In Me.txtFields
        PrimaryCLS.fields(i) = oText.Text
        i = i + 1
    Next
    mbAddNewFlag = False
    PrimaryCLS.Update
    Call bloquear(False)
    SetButtons True
    Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
    Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError

```

```

PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:
MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
On Error GoTo GoNextError
PrimaryCLS.MoveNext
Exit Sub
GoNextError:
MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub

```

frmeliminart

```

Private Sub cmdaceptar_Click()
If Trim(List1.Text) <> "" Then
smotivo = Me.List1.Text
Unload Me
Else
MsgBox "No hay motivos ", vbInformation, "Mensaje de Sistema"
End If
End Sub
Private Sub cmdcancelar_Click()
Unload Me
End Sub
Private Sub Form_Load()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim v_msgbox As Integer
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
'Empieza transacción
vquery = "select codigo,descripcion from estado-grupo where codigo<>'A'"
vquery = vquery & " order by descripcion"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
While Not rs.EOF

```

```

Me.List1.AddItem rs.fields(0) & " " & rs.fields(1)
rs.MoveNext
Wend
rs.Close
db.Close
smotivo = Empty
Exit Sub
AddErr:
MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub
Private Sub List1_DblClick()
If Trim(List1.Text) <> "" Then
smotivo = Me.List1.Text
Unload Me
End If
End Sub

```

frmempleado

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.3
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.65
DataGrid1.Columns(2).Width = 0
DataGrid1.Columns(3).Width = 0
DataGrid1.Columns(4).Width = 0
DataGrid1.Columns(5).Width = 0
DataGrid1.Columns(6).Width = 0
DataGrid1.Columns(7).Width = 0
DataGrid1.Columns(8).Width = 0
DataGrid1.Columns(0).Caption = "COD"
DataGrid1.Columns(1).Caption = "NOMBRE"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
cboestadocivil.Enabled = valor
End Sub
Private Sub cargartexto()
Dim i As Long
Dim j As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocanton.ConnectionString = ruta-base
i = i + 1
Next

```

```

For j = 0 To cboestadocivil.ListCount - 1
    If (Mid(cboestadocivil.List(j), 1, 1) = Trim(txtFields(3).Text)) Then
        cboestadocivil.ListIndex = j
    Exit For
End If
Next j
End Sub
Private Sub cboestadocivil_LostFocus()
    If (cboestadocivil.ListIndex > -1) Then
        txtFields(3).Text = Mid(cboestadocivil.Text, 1, 1)
    End If
End Sub
Private Sub Form_Load()
    vquery = "select * from empleado Order by Nombre"
    Set PrimaryCLS = New clsconton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub

```

```

End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
On Error GoTo AddErr
i = 0
For Each oText In Me.txtFields
oText.Text = Empty
i = i + 1
Next
Call bloquear(True)
lblStatus.Caption = "Agregar registro"
mbAddNewFlag = True
SetButtons False
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
PrimaryCLS.Delete
Call cargartexto
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr
lblStatus.Caption = "Modificar registro"
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
Call cargartexto
cmbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr

```



```

'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
  If Trim(oText.Text) = Empty Then
    oText.SetFocus
    MsgBox "Campo vacío no se puede grabar", vbCritical
    Exit Sub
  End If
Next
vquery = "select count(*) from empleado where codigo="" & Me.txtFields(0) & """
If duplicado And mbAddNewFlag = True Then
  txtFields(0).SetFocus
  MsgBox "Registro duplicado", vbCritical
  Exit Sub
End If
FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
  PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
  PrimaryCLS.fields(i) = oText.Text
  i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
SetButtons True
Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
  Unload Me
End Sub
Private Sub cmdFirst_Click()
  On Error GoTo GoFirstError
  PrimaryCLS.MoveFirst
  mbDataChanged = False
  Exit Sub
GoFirstError:
  MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
  On Error GoTo GoLastError
  PrimaryCLS.MoveLast
  mbDataChanged = False
  Exit Sub
GoLastError:
  MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
  On Error GoTo GoNextError
  PrimaryCLS.MoveNext
  Exit Sub
GoNextError:
  MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()

```



```

On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 1 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0
End If
End If
End If
End Sub

```

frmestadogrupo

```

Private WithEvents PrimaryCLS As clscanton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.7
DataGrid1.Columns(2).Width = DataGrid1.Width * 0
DataGrid1.Columns(0).Caption = "CODIGO"
DataGrid1.Columns(1).Caption = "DESCIUPCION"
DataGrid1.Columns(2).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
For Each oText In Me.Optestado
oText.Enabled = valor
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0

```

```

For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocanton.ConnectionString = ruta_base
Select Case i
Case 2
If txtFields(i).Text = "A" Then
Optestado(0).Value = True
Else
Optestado(1).Value = True
End If
End Select
i = i + 1
Next
End Sub
Private Sub Adocanton_WillMove(ByVal adReason As ADODB.EventReasonEnum, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
End Sub
Private Sub Form_Load()
vquery = "selectcodigo,descripcion,estado from estado-grupo Order by descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub
Select Case KeyCode
Case vbKeyEscape
cmdClose_Click
Case vbKeyEnd
cmdLast_Click
Case vbKeyHome
cmdFirst_Click
Case vbKeyUp, vbKeyPageUp
If Shift = vbCtrlMask Then
cmdFirst_Click
Else
cmdPrevious_Click
End If
Case vbKeyDown, vbKeyPageDown
If Shift = vbCtrlMask Then
cmdLast_Click
Else
cmdNext_Click
End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
Screen.MousePointer = vbDefault
End Sub

```



```

Private Sub PrimaryCLS_MoveComplete()
'Esto mostrará la posición de registro actual para este Recordset
lblStatus.Caption = "Registro: " & CStr(PrimaryCLS.AbsolutePosition)
If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
    Call cargartexto
End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
On Error GoTo AddErr
i = 0
For Each oText In Me.txtFields
    oText.Text = Empty
    i = i + 1
Next
Call bloquear(True)
lblStatus.Caption = "Agregar registro"
mbAddNewFlag = True
SetButtons False
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
PrimaryCLS.Delete
Call cargartexto
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr
lblStatus.Caption = "Modificar registro"
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
Call cargartexto

```

```

cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'Asignar el valor por default del radio button al texto
If Optestado(i).Value = True Then
txtFields(2).Text = "A"
Else
txtFields(2).Text = "I"
End If
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from estado_grupo where codigo=" & Me.txtFields(0) & ""
If duplicado And mbAddNewFlag = True Then
txtFields(0).SetFocus
MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
PrimaryCLS.fields(i) = oText.Text
i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
PrimaryCLS.MoveFirst
mbDataChanged = False
Exit Sub
GoFirstError:
MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
On Error GoTo GoLastError
PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:

```

```

    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdRefresh.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub

```

frmformapago

```

Private WithEvents PrimaryCLS As clscanton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.7
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0
    DataGrid1.Columns(0).Caption = "CODIGO"
    DataGrid1.Columns(1).Caption = "DESCRIPCION"
    DataGrid1.Columns(2).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
    Dim i As Long
    i = 0
    For Each oText In Me.txtFields
        oText.Enabled = valor
        i = i + 1
    Next
    For Each oText In Me.Optestado
        oText.Enabled = valor
    Next
End Sub
Private Sub cargartexto()
    Dim i As Long
    Dim a(1) As String

```

```

i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adocanton.ConnectionString = ruta-base
Select Case i
Case 2
If txtFields(i).Text = "A" Then
Optestado(0).Value = True
Else
Optestado(1).Value = True
End If
End Select
i = i + 1
Next
End Sub
Private Sub Form_Load()
vquery = "select codigo,descripcion,estado from formaqago Order by descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub
Select Case KeyCode
Case vbKeyEscape
cmdClose_Click
Case vbKeyEnd
cmdLast_Click
Case vbKeyHome
cmdFirst_Click
Case vbKeyUp, vbKeyPageUp
If Shift = vbCtrlMask Then
cmdFirst_Click
Else
cmdPrevious_Click
End If
Case vbKeyDown, vbKeyPageDown
If Shift = vbCtrlMask Then
cmdLast_Click
Else
cmdNext_Click
End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
'Esto mostrará la posición de registro actual para este Recordset

```

```

lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
    Call cargartexto
End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    mbEditFlag = True
    Call bloquear(True)
    SetButtons False
Exit Sub
EditErr:
    MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
    On Error Resume Next
    mbAddNewFlag = False
    PrimaryCLS.Cancel
    PrimaryCLS.Requery
    Call FormatoGrid
    SetButtons True
    Call bloquear(False)
    Call cargartexto
    cmdbuscarprov.Visible = False
End Sub

```



```

Private Sub cmdUpdate_Click()
Dim i As Long
  On Error GoTo UpdateErr
  'Asignar el valor por default del radio button al texto
  If Optestado(i).Value = True Then
    txtFields(2).Text = "A"
  Else
    txtFields(2).Text = "I"
  End If
  'Verificando que los campos no esten vacíos
  For Each oText In Me.txtFields
    If Trim(oText.Text) = Empty Then
      oText.SetFocus
      MsgBox "Campo vacío no se puede grabar", vbCritical
      Exit Sub
    End If
  Next
  vquery = "select count(*) from forma_pago where codigo=" & Me.txtFields(0) & ""
  If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
    Exit Sub
  End If
  'FIN DE VERIFICACIONES
  'graba registro nuevo
  If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
  End If
  i = 0
  For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
  Next
  mbAddNewFlag = False
  PrimaryCLS.Update
  Call bloquear(False)
  SetButtons True
  Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
  Unload Me
End Sub
Private Sub cmdFirst_Click()
  On Error GoTo GoFirstError
  PrimaryCLS.MoveFirst
  mbDataChanged = False
  Exit Sub
GoFirstError:
  MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
  On Error GoTo GoLastError
  PrimaryCLS.MoveLast
  mbDataChanged = False
  Exit Sub
GoLastError:
  MsgBox Err.Description
End Sub

```

```

Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdRefresh.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub

```

frmfoto

```

Private Sub cmdcerrar_Click()
    Unload Me
End Sub

```

frmgrupo

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim fotoruta As String
Public Sub eliminacion()
    Dim rs As ADODB.Recordset
    Dim db As ADODB.Connection
    Dim scodigotipo As String
    Dim sactividad As String
    Dim lcodigo As Long
    On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta-base
    Me.txtfields(12).Text = Format(Date, "dd/mm/yyyy")
    mebfields(2).Mask = ""
    mebfields(2).Text = txtfields(12).Text
    'tomo codigos de los combos
    scodigotipo = Mid(cmbfields(1).Text, 1, InStr(1, cmbfields(1).Text, " ") - 1)
    sactividad = Mid(cmbfields(2).Text, 1, InStr(1, cmbfields(2).Text, " ") - 1)

```

```

'Empieza transacción
db.BeginTrans
'Código siguiente
vquery = "select codigo-estado from bitacora-grupo order by codigo-estado "
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not rs.EOF Then
rs.MoveLast
If Not IsNull(rs.fields(0)) Then
lcodigo = rs.fields(0) + 1
Else
lcodigo = 1
End If
Else
lcodigo = 1
End If
rs.Close
'insertar nueva clausura
vquery = "insert into bitacora_grupo(codigo_grupo,codigo_tipo_actividad,"
vquery = vquery & "codigo_actividad,codigo_estado,fecha,observacion,"
vquery = vquery & "estado) VALUES ("
vquery = vquery & "" & Trim(txtfields(0)) & "," & Trim(scodigotipo) & ""
vquery = vquery & "," & Trim(sactividad) & "," & Trim(Str(lcodigo))
vquery = vquery & "," & Trim(txtfields(12)) & "," & Trim(Mid(smotivo, InStr(1, smotivo, " ") + 1) &
", " & Trim(Mid(smotivo, 1, InStr(1, smotivo, " ") - 1) & ""
'Manda a ejecutar el query
db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub 'eliminacion
Public Sub levantaclausura()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim scodigotipo As String
Dim sactividad As String
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
Me.txtfields(14).Text = Empty
Me.txtfields(12).Text = Empty
'tomo codigos de los combos
scodigotipo = Mid(cmbfields(1).Text, 1, InStr(1, cmbfields(1).Text, " ") - 1)
sactividad = Mid(cmbfields(2).Text, 1, InStr(1, cmbfields(2).Text, " ") - 1)
'Empieza transacción
db.BeginTrans
'actualizo
vquery = "update grupo-turistico set fecha_clausura=NULL"
vquery = vquery & ",Observacion=" "
vquery = vquery & ",estado='A' "
vquery = vquery & "where codigo=" & Trim(txtfields(0)) & "" and "
vquery = vquery & "codigo-tipo-actividad=" & Trim(scodigotipo) & "" and "
vquery = vquery & "codigo_actividad=" & Trim(sactividad) & ""

```

```

'Manda a ejecutar el query
db.Execute vquery
'Código siguiente
vquery = "select codigo-estado from bitacora_grupo order by codigo-estado"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not rs.EOF Then
    rs.MoveLast
    If Not IsNull(rs.fields(0)) Then
        lcodigo = rs.fields(0) + 1
    Else
        lcodigo = 1
    End If
Else
    lcodigo = 1
End If
rs.Close
'insertar nueva clausura
vquery = "insert into bitacora_grupo(codigo_grupo,codigo_tipo_actividad,"
vquery = vquery & "codigo_actividad,codigo_estado,fecha,"
vquery = vquery & "estado) VALUES ("
vquery = vquery & """" & Trim(txtfields(0)) & """, "" & Trim(scodigotipo) & """"
vquery = vquery & """, "" & Trim(sactividad) & """, "" & Trim(Str(lcodigo))
vquery = vquery & """, "" & Trim(Format(Date, "dd/mm/yyyy")) & """,'L'"
'Manda a ejecutar el query
db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Sub 'levantaclausura
Public Sub clausura()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim scodigotipo As String
Dim sactiviáad As String
Dim lcodigo As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
Me.txtfields(14).Text = smotivo
Me.txtfields(12).Text = Format(Date, "dd/mm/yyyy")
mebfields(2).Mask = ""
mebfields(2).Text = txtfields(12).Text
'tomo codigos de los combos
scodigotipo = Mid(cmbfields(1).Text, 1, InStr(1, cmbfields(1).Text, " ") - 1)
sactividad = Mid(cmbfields(2).Text, 1, InStr(1, cmbfields(2).Text, " ") - 1)
'Empieza transacción
db.BeginTrans
'actualizo
vquery = "update grupo-turistico set fecha-clausura="" & Trim(txtfields(12))& """"
vquery = vquery & ",Observacion="" & Trim(txtfields(14))& """"
vquery = vquery & " where código="" & Trim(txtfields(0)) & "" and "
vquery = vquery & " codigo_tipo_actividad="" & Trim(scodigotipo) & "" and "
vquery = vquery & " codigo_actividad="" & Trim(sactividad) & """"

```

```

'Manda a ejecutar el query
db.Execute vquery
'Código siguiente
vquery = "select codigo-estado from bitacora-grupo order by codigo-estado "
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not rs.EOF Then
    rs.MoveLast
    If Not IsNull(rs.fields(0)) Then
        lcodigo =rs.fields(0) + 1
    Else
        lcodigo = 1
    End If
Else
    lcodigo = 1
End If
rs.Close
'insertar nueva clausura
vquery = "insert into bitacora_grupo(codigo_grupo,codigo_tipo_actividad,"
vquery = vquery & "codigo_actividad,codigo_estado,fecha,observacion,"
vquery = vquery & "estado) VALUES ("
vquery = vquery & """" & Trim(txtfields(0)) & """, "" & Trim(scodigotipo) & """"
vquery = vquery & "" & Trim(sactividad) & """, "" & Trim(Str(lcodigo))
vquery = vquery & "" & Trim(txtfields( 12))& """, "" & Trim(txtfields( 14))& ""', 'C')"
'Manda a ejecutar el query
db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub 'clausura
Public Sub Ejecutarquery()
    Dim rs As ADODB.Recordset
    Dim db As ADODB.Connection
    Dim vtemp As String
    Dim j As Integer
    'On Error Go To AddErr
    If Trim(txtfields(0)) <> "" And Trim(cmbfields(1).Text) <> "" And Trim(cmbfields(1).Text) <> ""
Then
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta_base
    SetButtons "EXECUTE"
    'No permite actualizar clave primaria
    'Empieza transacción
    'db.BeginTrans
    'busco el grupo que quiero consultar
    vquery = "select count(*) from grupo-turistico "
    vquery = vquery & " where codigo=" & Me.txtfields(0) & """"
    vquery = vquery & " and codigo_tipo_actividad=" & cmbfields(1).Text & """"
    vquery = vquery & " and codigo_actividad=" & cmbfields(2).Text & """"
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    If rs.fields(0) = 0 Then
        MsgBox "El Grupo que esta consultando NO EXISTE", vbCritical, "Mensaje del Sistema
EDEN"
    End If
Exit Sub

```

```

End If
rs.Close
'recupero datos del grupo
vquery = "select propietario,representante,razon_social,codigo_provincia,codigo_canton,"
vquery = vquery &
"codigo_parroquia,ubicacion,telefono,email,fax,fecha_operacion,fecha_ingreso,"
vquery = vquery &
"fecha_clausura,folio,estado,nombre,codigo,codigo_tipo_actividad,codigo_actividad "
vquery = vquery &
",identificacion,codigo_categoria,tipo_local,Observacion,codigo_sector,valor,MSLINK from
grupo-turistico"
vquery = vquery & " where codigo=" & Me.txtfields(0) & ""
vquery = vquery & " and codigo_tipo_actividad=" & cmbfields(1).Text & ""
vquery = vquery & " and codigo_actividad=" & cmbfields(2).Text & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not IsNull(rs.fields(0)) Then
txtfields(2).Text = rs.fields(0)
End If
If Not IsNull(rs.fields(1)) Then
txtfields(3).Text = rs.fields(1)
End If
If Not IsNull(rs.fields(2)) Then
txtfields(4).Text = rs.fields(2)
End If
If Not IsNull(rs.fields(13)) Then
txtfields(5).Text = rs.fields(13)
End If
If Not IsNull(rs.fields(11)) Then
txtfields(6).Text = rs.fields(11)
End If
mebfields(0).Mask = ""
mebfields(0).Text = txtfields(6).Text
If Not IsNull(rs.fields(10)) Then
txtfields(7).Text = rs.fields(10)
End If
mebfields(1).Mask = ""
mebfields(1).Text = txtfields(7).Text
'habilita boton de clausura
cmdclausura.Visible = True
If Not IsNull(rs.fields(12)) Then
txtfields(12).Text = rs.fields(12)
Me.cmdclausura.Caption = "Levantar Clausura"
mebfields(2).Mask = ""
mebfields(2).Text = txtfields(12).Text
Else
Me.cmdclausura.Caption = "Clausurar"
End If
If Not IsNull(rs.fields(7)) Then
txtfields(8).Text = rs.fields(7)
End If
If Not IsNull(rs.fields(9)) Then
txtfields(9).Text = rs.fields(9)
End If
If Not IsNull(rs.fields(8)) Then
txtfields(10).Text = rs.fields(8)
End If
If Not IsNull(rs.fields(6)) Then
txtfields(11).Text = rs.fields(6)
End If
If Not IsNull(rs.fields(15)) Then

```

```

    txtfields(1).Text = rs.fields(15)
End If
If Not IsNull(rs.fields(19)) Then
    txtfields(13).Text = rs.fields(19)
End If
If Not IsNull(rs.fields(24)) Then
    txtfields(15).Text = rs.fields(24)
End If
If Not IsNull(rs.fields(25)) Then
    txtfields(16).Text = rs.fields(25)
End If
'Observación
If Not IsNull(rs.fields(22)) Then
    txtfields(14).Text = rs.fields(22)
End If
Me.cmbfields(0).Clear
vquery = "select codigo,descripcion from estado_grupo"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(0), 1)
For j = 0 To cmbfields(0).ListCount - 1
    If (rs.fields(14) = Mid(cmbfields(0).List(j), 1, InStr(1, cmbfields(0).List(j), " ") - 1)) Then
        cmbfields(0).ListIndex = j
    Exit For
End For
End If
Next j
'Llena tipo actividad
Me.cmbfields(1).Clear
vquery = "select codigo,descripcion from tipo_actividad order by descripcion"
Call llenacombo(Me.cmbfields(1), 1)
For j = 0 To cmbfields(1).ListCount - 1
    If (rs.fields(17) = Mid(cmbfields(1).List(j), 1, InStr(1, cmbfields(1).List(j), " ") - 1)) Then
        cmbfields(1).ListIndex = j
    Exit For
End For
Next j
'Llena provincia
Me.cmbfields(3).Clear
vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
Call llenacombo(Me.cmbfields(3), 1)
For j = 0 To cmbfields(3).ListCount - 1
    If (rs.fields(3) = Val(Mid(cmbfields(3).List(j), 1, InStr(1, cmbfields(3).List(j), " ") - 1))) Then
        cmbfields(3).ListIndex = j
    Exit For
End For
Next j
'Llena actividad
For j = 0 To cmbfields(2).ListCount - 1
    If (rs.fields(18) = Val(Mid(cmbfields(2).List(j), 1, InStr(1, cmbfields(2).List(j), " ") - 1))) Then
        cmbfields(2).ListIndex = j
    Exit For
End If
Next j
'Llena categoria
For j = 0 To cmbfields(6).ListCount - 1
    If (rs.fields(20) = Val(Mid(cmbfields(6).List(j), 1, InStr(1, cmbfields(6).List(j), " ") - 1))) Then
        cmbfields(6).ListIndex = j
    Exit For
End If
Next j
'llena tipo local

```

```

cmbfields(8).Clear
vquery = "select codigo,descripcion from tipo_local"
vquery = vquery & " where estado='A'"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(8), 3)
'llena tipo local
For j = 0 To cmbfields(8).ListCount - 1
  If (rs.fields(2) = Mid(cmbfields(8).List(j), 1, InStr(1, cmbfields(8).List(j), " ") - 1)) Then
    cmbfields(8).ListIndex = j
  Exit For
End If
Next j
'llena canton
For j = 0 To cmbfields(4).ListCount - 1
  If (rs.fields(4) = Val(Mid(cmbfields(4).List(j), 1, InStr(1, cmbfields(4).List(j), " ") - 1))) Then
    cmbfields(4).ListIndex = j
  Exit For
End If
Next j
'llena parroquia
For j = 0 To cmbfields(5).ListCount - 1
  If (rs.fields(5) = Val(Mid(cmbfields(5).List(j), 1, InStr(1, cmbfields(5).List(j), " ") - 1))) Then
    cmbfields(5).ListIndex = j
  Exit For
End If
Next j
'llena sector
If Not IsNull(rs.fields(23)) Then
  For j = 0 To cmbfields(7).ListCount - 1
    If (rs.fields(23) = Val(Mid(cmbfields(7).List(j), 1, InStr(1, cmbfields(7).List(j), " ") - 1)))
Then
      cmbfields(7).ListIndex = j
    Exit For
  End If
Next j
End If
'llena estado
For j = 0 To cmbfields(0).ListCount - 1
  If (rs.fields(15) = Mid(cmbfields(0).List(j), 1, InStr(1, cmbfields(0).List(j), " ") - 1)) Then
    cmbfields(0).ListIndex = j
  Exit For
End If
Next j
rs.Close
End If
Call bloquear(True)
txtfields(0).Enabled = False
cmbfields(1).Enabled = False
cmbfields(2).Enabled = False
txtfields(15).Enabled = False
txtfields(16).Enabled = False
'botones del grupo
cmdinstalacion.Enabled = True
Cmdservicio.Enabled = True
cmdfoto.Enabled = True
cmdimprimir.Enabled = True
Cmdmapa.Enabled = True
mbAddNewFlag = False
End Sub 'ejecutarquery

```


Private Sub SetButtons(accion As String)

Select Case accion

Case "ADD"

cmdadd.Visible = False
cmdconsultar.Visible = False
cmdcerrar.Visible = False
cmdcancel.Visible = True
cmdupdate.Visible = True
cmdeliminar.Visible = False
cmdejecutar.Visible = False
cmdclausura.Visible = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False
'deshabilito mslink
txtfields(16).Enabled = False

Case "QUERY"

cmdadd.Visible = False
cmdconsultar.Visible = False
cmdcerrar.Visible = False
cmdcancel.Visible = True
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = True
cmdclausura.Visible = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False

Case "EXECUTE"

cmdadd.Visible = False
cmdconsultar.Visible = False
cmdcerrar.Visible = False
cmdcancel.Visible = True
cmdupdate.Visible = True
cmdeliminar.Visible = True
cmdejecutar.Visible = False
cmdclausura.Visible = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False

Case "DEL"

cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
cmdinstalacion.Enabled = False
Cmdservicio.Enabled = False
cmdfoto.Enabled = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False

Case "CANCEL"

cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
cmdinstalacion.Enabled = False
Cmdservicio.Enabled = False

```

cmdfoto.Enabled = False
cmdclausura.Visible = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False
Case "UPDATE"
cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
cmdinstalacion.Enabled = False
Cmdservicio.Enabled = False
cmdfoto.Enabled = False
cmdimprimir.Enabled = False
Cmdmapa.Enabled = False
'deshabilita mslink y valor
txtfields(16).Enabled = False
txtfields(15).Enabled = False
End Select
End Sub
Private Sub bloquear(valor As Boolean)
'bloquea los textos
For Each oText In Me.txtfields
oText.Enabled = valor
Next
'bloquea los combos
For Each oText In Me.cmbfields
oText.Enabled = valor
Next
'bloquea las mascaras
For Each oText In Me.mebfields
oText.Enabled = valor
Next
End Sub
Private Sub Adodc2_MoveComplete(ByVal adReason As ADODB.EventReasonEnum, ByVal
pError As ADODB.Error, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As
ADODB.Recordset)
Dim i As Integer
vcodigo = ""
If Not Adodc2.Recordset.EOF And Not Adodc2.Recordset.BOF Then
frmgrupo.txtfields(0).Text = Me.Adodc2.Recordset.fields(0)
frmgrupo.cmbfields(1).Text = Me.Adodc2.Recordset.fields(1)
frmgrupo.cmbfields(2).Text = Me.Adodc2.Recordset.fields(2)
Call frmgrupo.Ejecutarquery
End If
End Sub
Private Sub Cmbfields_Click(Index As Integer)
Dim scanton As String
Dim sprov As String
Select Case Index
Case 0
If Trim(cmbfields(0).Text) <> "" Then
If Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") - 1) = "I" Then
txtfields(12) = Format(Date, "dd/mm/yyyy")
End If
End If
Case 1
If Trim(cmbfields(1).Text) <> "" Then

```

```

    cmbfields(2).Clear
    vquery = "select codigo,descripcion from actividad"
    vquery = vquery & " Where codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & ""
    vquery = vquery & " order by descripcion"
    End If
    Call llenacombo(Me.cmbfields(2), 2)
Case 3
    If Trim(cmbfields(3).Text) <> "" Then
        cmbfields(4).Clear
        cmbfields(5).Clear
        vquery = "select codigo_canton,descripcion fi-om canton"
        vquery = vquery & " Where codigo_provincia=" & Mid(cmbfields(3).Text, 1, InStr(1,
cmbfields(3).Text, " ") - 1)
        vquery = vquery & " order by descripcion"
    End If
    Call llenacombo(Me.cmbfields(4), 2)
Case 4
    If Trim(cmbfields(3).Text) <> "" And Trim(cmbfields(4).Text) <> "" Then
        cmbfields(5).Clear
        scanton = Mid(cmbfields(4).Text, 1, InStr(1, cmbfields(4).Text, " ") - 1)
        sprov = Mid(cmbfields(3).Text, 1, InStr(1, cmbfields(3).Text, " ") - 1)
        vquery = "select codigo_parroquia,descripcion from parroquia "
        vquery = vquery & " Where codigo_canton=" & scanton
        vquery = vquery & " and codigo_provincia=" & sprov
        vquery = vquery & " order by descripcion"
    End If
    Call llenacombo(Me.cmbfields(5), 3)
Case 5
    If Trim(cmbfields(4).Text) <> "" And Trim(cmbfields(3).Text) <> "" And
Trim(cmbfields(5).Text) <> "" Then
        cmbfields(7).Clear
        scanton = Mid(cmbfields(4).Text, 1, InStr(1, cmbfields(4).Text, " ") - 1)
        sprov = Mid(cmbfields(3).Text, 1, InStr(1, cmbfields(3).Text, " ") - 1)
        sparro = Mid(cmbfields(5).Text, 1, InStr(1, cmbfields(5).Text, " ") - 1)
        vquery = "select codigo,descripcion from sector"
        vquery = vquery & " Where codigo_canton=" & scanton
        vquery = vquery & " and codigo_provincia=" & sprov
        vquery = vquery & " and codigo_parroquia=" & sparro
        vquery = vquery & " order by descripcion"
    End If
    Call llenacombo(Me.cmbfields(7), 3)
End Select
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtfields
        oText.Text = Empty
    Next
    For Each oText In Me.mebfields
        oText.Mask = ""
        oText.Text = ""
        oText.Mask = "##-??-####"
    Next
    'llena estado
    Me.cmbfields(0).Clear
    vquery = "select codigo,descripcion fi-om estado_grupo"
    vquery = vquery & " order by descripcion"

```

```

Call llenacombo(Me.cmbfields(0), 1)
fotoruta = Empty
'Llena tipo actividad
Me.cmbfields(1).Clear
vquery = "select codigo,descripcion fi-om tipo-actividad order by descripcion"
Call llenacombo(Me.cmbfields(1), 1)
'Llena provincia
Me.cmbfields(3).Clear
vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
Call llenacombo(Me.cmbfields(3), 1)
'Llena categoria
cmbfields(6).Clear
vquery = "select codigo,descripcion fi-om categoria"
vquery = vquery & " where estado='A'"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(6), 3)
'Llena tipo local
cmbfields(8).Clear
vquery = "select codigo,descripcion from tipo_local"
vquery = vquery & " where estado='A'"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(8), 3)
Call bloquear(True)
mbAddNewFlag = True
SetButtons "ADD"
txtfields(0).SetFocus
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
For Each oText In Me.txtfields
oText.Text = Empty
Next
For Each oText In Me.cmbfields
oText.Text = Empty
Next
For Each oText In Me.mebfields
oText.Mask = ""
oText.Text = ""
oText.Mask = "##-??-#####"
Next
Call bloquear(False)
SetButtons "CANCEL"
'Adodc2.Recordset.Close
End Sub
Private Sub cmdcerrar_Click()
Unload Me
End Sub
Private Sub cmdclausura_Click()
If Trim(Me.cmdclausura.Caption) = "Clausurar" Then
frmclausura.Show vbModal
If smotivo <> Empty Then
Call clausura
Me.cmdclausura.Caption = "Levantar Clausura"
End If
Else
Call levantaclausura
mebfields(2).Mask = ""
mebfields(2).Text = ""

```

```

    Me.cmdclausura.Caption = "Clausurar"
End If
End Sub
Private Sub cmdconsultar_Click()
    Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtfields
        oText.Text = Empty
    Next
    For Each oText In Me.mebfields
        oText.Mask = ""
        oText.Text = ""
        oText.Mask = "##-??-####"
    Next
'lena estado
Me.cmbfields(0).Clear
vquery = "select codigo,descripcion from estado_grupo"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(0), 1)
'lena categoria
cmbfields(6).Clear
vquery = "select codigo,descripcion from categoria"
vquery = vquery & " where estado='A'"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(6), 3)
'lena tipo actividad
Me.cmbfields(1).Clear
vquery = "select codigo,descripcion from tipo-actividad"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(1), 1)
'lena provincia
Me.cmbfields(3).Clear
vquery = "select codigo_provincia,descripcion from provincia"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(3), 1)
'lena tipo local
cmbfields(8).Clear
vquery = "select codigo,descripcion from tipo-local"
vquery = vquery & " where estado='A'"
vquery = vquery & " order by descripcion"
Call llenacombo(Me.cmbfields(8), 3)
Call bloquear(True)
mbAddNewFlag = True
SetButtons "QUERY"
txtfields(0).SetFocus
txtfields(14).Enabled = False
Exit Sub
AddErr:
    MsgBox Err.Description
End Su
Private Sub cmdejecutar_Click()
    Dim rs As ADODB.Recordset
    Dim db As ADODB.Connection
    Dim vtemp As String
    SetButtons "QUERY"
' muestra los grupos en una ventana para ayudar a la consulta
vquery = "select g.codigo as CODIGO,g.codigo_tipo_actividad as
TIPO_ACTIVIDAD,g.codigo_actividad as ACTIVIDAD,g.nombre as NOMBRE,ta.descripcion as
DESCRIPCION_TIPO,a.descripcion as DESCRIPCION_ACTIVIDAD "

```

```

vquery = vquery & "from grupo-turistico g,tipo_actividad ta,actividad a "
vquery = vquery & "where g.codigo_tipo_actividad = ta.codigo and "
vquery = vquery & "g.codigo_tipo_actividad = a.codigo_tipo_actividad and
g.codigo_actividad=a.codigo"
'Añadir filtros
'Estado
If Trim(cmbfields(0).Text) <> "" Then
  vquery = vquery & " and g.estado=" & Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") -
1) & ""
End If
'Registro
If Trim(txtfields(0).Text) <> "" Then
  vquery = vquery & " and g.codigo=" & txtfields(0).Text & ""
End If
'Nombre de Grupo
If Trim(txtfields(1).Text) <> "" Then
  vquery = vquery & " and g.nombre like '%" & txtfields(1).Text & "%"
End If
'Nombre de propietario
If Trim(txtfields(2).Text) <> "" Then
  vquery = vquery & " and g.propietario like '%" & txtfields(2).Text & "%"
End If
'Nombre de representante legal
If Trim(txtfields(3).Text) <> "" Then
  vquery = vquery & " and g.representante like 'Yo'" & txtfields(3).Text & "%"
End If
'Razón Social
If Trim(txtfields(4).Text) <> "" Then
  vquery = vquery & " and g.razon_social like '%" & txtfields(4).Text & "Yo"
End If
'provincia
If Trim(cmbfields(3).Text) <> "" Then
  vquery = vquery & " and g.codigo_provincia=" & Mid(cmbfields(3).Text, 1, InStr(1,
cmbfields(3).Text, " ") - 1)
End If
'cantón
If Trim(cmbfields(4).Text) <> "" Then
  vquery = vquery & " and g.codigo_canton=" & Mid(cmbfields(4).Text, 1, InStr(1,
cmbfields(4).Text, " ") - 1)
End If
'parroquia
If Trim(cmbfields(5).Text) <> "" Then
  vquery = vquery & " and g.codigo_parroquia=" & Mid(cmbfields(5).Text, 1, InStr(1,
cmbfields(5).Text, " ") - 1)
End If
'ubicación
If Trim(txtfields(11).Text) <> "" Then
  vquery = vquery & " and g.ubicacion like '%" & txtfields(11).Text & "%"
End If
'telefono
If Trim(txtfields(8).Text) <> "" Then
  vquery = vquery & " and g.telefono like '%" & txtfields(8).Text & "%"
End If
'email
If Trim(txtfields(10).Text) <> "" Then
  vquery = vquery & " and g.email like '%" & txtfields(10).Text & "%"
End If
'fax
If Trim(txtfields(9).Text) <> "" Then
  vquery = vquery & " and g.fax like '%" & txtfields(9).Text & "%"

```

```

End If
'folio
If Trim(txtfields(5).Text) <> "" Then
    vquery = vquery & " and g.folio like '%" & txtfields(5).Text & "%"
End If
'tipo actividad
If Trim(cmbfields(1).Text) <> "" Then
    vquery = vquery & " and g.codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & ""
End If
'tipo actividad
If Trim(cmbfields(2).Text) <> "" Then
    vquery = vquery & " and g.codigo_actividad=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & ""
End If
'tipo local
If Trim(cmbfields(8).Text) <> "" Then
    vquery = vquery & " and g.tipo_local=" & Mid(cmbfields(8).Text, 1, InStr(1, cmbfields(8).Text, "
") - 1) & ""
End If
'fecha_ingreso
If Trim(txtfields(6).Text) <> "" Then
    vquery = vquery & " and g.fecha_ingreso=Cdate('" & (txtfields(6).Text) & "')"
End If
'fecha-operacion
If Trim(txtfields(7).Text) <> "" Then
    vquery = vquery & " and g.fecha_operacion=Cdate('" & txtfields(7).Text & "')"
End If
'fecha_clausura
If Trim(txtfields(12).Text) <> "" Then
    vquery = vquery & " and g.fecha_clausura=Cdate('" & txtfields(12).Text & "')"
End If
'identificación
If Trim(txtfields(13).Text) <> "" Then
    vquery = vquery & " and g.identificacion like '%" & txtfields(13).Text & "%"
End If
'sector
If Trim(cmbfields(7).Text) <> "" Then
    vquery = vquery & " and g.codigo_sector=" & Mid(cmbfields(7).Text, 1, InStr(1,
cmbfields(7).Text, " ") - 1)
End If
'orden de presentacion
vquery = vquery & " order by g.nombre"
frmconsultar.Col = 0
frmconsultar.grupo_punto = "Grupo"
Set Adodc2.Recordset = frmconsultar.Adoconsultar.Recordset
frmconsultar.Show
End Sub
Private Sub cmdeliminar_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim v_msgbox As Integer
Dim j As Integer
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
v_msgbox = MsgBox("Esta seguro de ELIMINAR este Grupo Turístico", vbOKCancel, "Mensaje
del Sistema EDEN")

```

```

If (v_msgbox = vbCancel) Then
    Exit Sub
End If
db.Open ruta_base
'Empieza transacción
db.BeginTrans
vquery = "select count(*) from grupo-turistico "
vquery = vquery & " where codigo=" & Me.txtfields(0) & ""
vquery = vquery & " and codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & ""
vquery = vquery & " and codigo_actividad=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.fields(0) = 0 Then
    MsgBox "El Grupo que quiere ELIMINAR NO EXISTE", vbCritical, "Mensaje del Sistema
EDEN"
    Exit Sub
End If
rs.Close
frmeliminar.Show vbModal
If smotivo = Empty Then
    Exit Sub
End If
cmdclausura.Visible = False
For j = 0 To cmbfields(0).ListCount - 1
    If (Mid(smotivo, 1, InStr(1, smotivo, " ") - 1) = Mid(cmbfields(0).List(j), 1, InStr(1,
cmbfields(0).List(j), " ") - 1)) Then
        cmbfields(0).ListIndex = j
    Exit For
End If
Next j
txtfields(12) = Format(Date, "dd/mm/yyyy")
vquery = "update grupo-turistico set "
vquery = vquery & "estado=" & Mid(smotivo, 1, InStr(1, smotivo, " ") - 1) & ""
vquery = vquery & ",fecha-clausura=" & Trim(txtfields(12)) & ""
vquery = vquery & " where codigo=" & Me.txtfields(0) & ""
vquery = vquery & " and codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & ""
vquery = vquery & " and codigo_actividad=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & ""
'Manda a ejecutar el query
db.Execute vquery
Call eliminacion
'Termina la transacción
db.CommitTrans
db.Close
For Each oText In Me.txtfields
    oText.Text = Empty
Next
For Each oText In Me.cmbfields
    oText.Text = Empty
Next
Call bloquear(False)
SetButtons "DEL"
Exit Sub
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Sub

```



```

Private Sub cmdfoto_Click()
On Error Resume Next ' Establece el tratamiento de errores.
Err.Clear
If Trim(txtfields(0).Text) <> "" And Trim(rutajpg) <> Empty Then
    fotoruta = Trim(rutajpg) & "\" & Trim(txtfields(0).Text) & ".jpg"
End If
If fotoruta <> Empty Then
    frmfoto.webfoto.Navigate fotoruta
    frmfoto.Show
Else
    MsgBox "No hay gráfica que coincida con la ruta", vbInformation, "Mensaje del Sistema"
End If
End Sub
Private Sub Cmdimprimir_Click()
    Me.CryReport.ReportFileName = rutarpt & "\grupo_turistico.rpt"
    Me.CryReport.PrintFileName = "grupo_turistico.rpt"
    vquery = "{grupo_turistico.codigo}=" & txtfields(0).Text & " and " & "
{grupo_turistico.codigo_tipo_actividad}=" & Mid(cmbfields(1).Text, 1, InStr(1, cmbfields(1).Text,
" ") - 1) & " and " & "{grupo_turistico.codigo_actividad}=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & ""
    Me.CryReport.SelectionFormula = vquery
    Me.CryReport.Action = True
End Sub
Private Sub cmdinstalacion_Click()
frmgrupoinstalacion.txtcodigo = txtfields(0)
frmgrupoinstalacion.txtgrupo = txtfields(1)
frmgrupoinstalacion.txttipoactiv = Me.cmbfields(1).Text
frmgrupoinstalacion.txtactividad = Me.cmbfields(2).Text
frmgrupoinstalacion.Show vbModal
End Sub
Private Sub Cmdmapa_Click()
    vopcion = 1
    frm_mapa.Show
End Sub
Private Sub Cmdservicio_Click()
frmgruposervicio.txtcodigo = txtfields(0)
frmgruposervicio.txtgrupo = txtfields(1)
frmgruposervicio.txttipoactiv = Me.cmbfields(1).Text
frmgruposervicio.txtactividad = Me.cmbfields(2).Text
frmgruposervicio.Show vbModal
End Sub
Private Sub cmdUpdate_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim scodigotipo As String
Dim sactividad As String
Dim sprovincia As String
Dim scanton As String
Dim sestado As String
Dim ssector As String
Dim sparroquia As String
Dim scategoria As String
Dim stipolocal As String
Dim esfecha
Dim vmslink As Double
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient

```

```

db.Open ruta-base
'Validación de textos o combos vacios
For Each oText In Me.txtfields
If (oText.Index >= 0 And oText.Index <= 4 Or oText.Index = 13 Or oText.Index = 15) Then
'campos requeridos
If Trim(oText) = "" Then
MsgBox "Favor llenar textos requeridos", vbInformation, "Mensaje del Sistema EDEN"
oText.SetFocus
Exit Sub
End If
End If
Next
For Each oText In Me.cmbfields
If (oText.Index >= 1 And oText.Index <= 5) Then 'campos requeridos
If Trim(oText.Text) = "" Then
MsgBox "Favor escoger un elemento de la lista", vbInformation, "Mensaje del Sistema EDEN"
oText.SetFocus
Exit Sub
End If
End If
Next
'tomo codigos de íos combos
scodigotipo = Mid(cmbfields(1).Text, 1, InStr(1, cmbfields(1).Text, " ") - 1)
sactividad = Mid(cmbfields(2).Text, 1, InStr(1, cmbfields(2).Text, " ") - 1)
sprovincia = Mid(cmbfields(3).Text, 1, InStr(1, cmbfields(3).Text, " ") - 1)
scanton = Mid(cmbfields(4).Text, 1, InStr(1, cmbfields(4).Text, " ") - 1)
If Trim(cmbfields(7).Text) <> "" Then
ssector = Mid(cmbfields(7).Text, 1, InStr(1, cmbfields(7).Text, " ") - 1)
Else
ssector = "0"
End If
sparroquia = Mid(cmbfields(5).Text, 1, InStr(1, cmbfields(5).Text, " ") - 1)
sestado = Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") - 1)
If Trim(cmbfields(6).Text) <> "" Then
scategoria = Mid(cmbfields(6).Text, 1, InStr(1, cmbfields(6).Text, " ") - 1)
End If
stipolocal = Mid(cmbfields(8).Text, 1, InStr(1, cmbfields(8).Text, " ") - 1)
'valido campos de fecha
If (IsDate(txtfields(6).Text) = False) Then
MsgBox "El formato de la Fecha de Registro es INCORRECTO", vbExclamation
Exit Sub
End If
If (IsDate(txtfields(7).Text) = False) Then
MsgBox "El formato de la Fecha de Operaciones es INCORRECTO", vbExclamation
Exit Sub
End If
If (IsDate(txtfields(12).Text) = False) Then
mebfields(2).Mask = ""
mebfields(2).Text = ""
txtfields(12).Text = mebfields(2).Text
mebfields(2).Mask = "##-??-#####"
End If
'Empieza transacción
db.BeginTrans
If mbAddNewFlag = True Then
'busco duplicados
vquery = "select count(*) from grupo-turistico "
vquery = vquery & " where código=" & Me.txtfields(0) & ""

```

```

vquery = vquery & " and codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & """"
vquery = vquery & " and codigo_actividad=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & """"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.fields(0) > 0 Then
    MsgBox "El Grupo ya existe", vbCritical, "Mensaje del Sistema"
    Exit Sub
End If
rs.Close
'obtengo nuevo mslink
vquery = "select max(MSLINK)+1 from grupo_turistico"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.fields(0) <> rs.EOF And rs.fields(0) <> rs.EOF And IsNull(rs.fields(0)) = False Then
    vmslink = rs.fields(0)
End If
rs.Close
vquery = "insert into grupo_turistico(codigo,codigo_tipo_actividad,"
vquery = vquery & "codigo_actividad,nombre,propietario,representante,"
vquery = vquery & "razon_social,codigo_provincia,codigo_canton,"
vquery = vquery & "codigo_parroquia,ubicacion,telefono,email,fax,"
vquery = vquery & "fecha_operacion,fecha_ingreso,"
If Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") - 1) = "I" Then
    vquery = vquery & "fecha-clausura,"
End If
vquery = vquery &
"folio,estado,tipo_local,codigo_categoria,codigo_sector,identificacion,Observacion,valor,MSLINK)
"
vquery = vquery & " VALUES("
vquery = vquery & """" & Trim(txtfields(0)) & """, "" & Trim(scodigotipo) & """"
vquery = vquery & """, "" & Trim(sactividad) & """, "" & Trim(txtfields(1)) & """"
vquery = vquery & """, "" & Trim(txtfields(2)) & """, "" & Trim(txtfields(3)) & """"
vquery = vquery & """, "" & Trim(txtfields(4)) & """, "" & Trim(sprovincia)
vquery = vquery & """, "" & Trim(scanton) & """, "" & Trim(sparroquia) & """, "" & Trim(txtfields(11)) &
""""
vquery = vquery & """, "" & Trim(txtfields(8)) & """, "" & Trim(txtfields(10)) & """"
vquery = vquery & """, "" & Trim(txtfields(9)) & """, "" & Trim(txtfields(7)) & """"
vquery = vquery & """, "" & Trim(txtfields(6)) & """"
If Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") - 1) = "I" Then
    vquery = vquery & """, "" & Trim(txtfields(12)) & """"
End If
vquery = vquery & """, "" & Trim(txtfields(5)) & """"
vquery = vquery & """, "" & Trim(estado) & """"
vquery = vquery & """, "" & Trim(stipolocal) & """"
vquery = vquery & """, "" & Trim(scategoria) & """"
vquery = vquery & """, "" & Trim(ssector)
vquery = vquery & """, "" & Trim(txtfields(13)) & """"
vquery = vquery & """, "" & Trim(txtfields(14)) & """"
vquery = vquery & """, "" & Trim(txtfields(15)) & """"
vquery = vquery & """, "" & vmslink & """"
vquery = vquery & ")"
Else
Hacer el query para actualizar
vquery = "select count(*) from grupo_turistico"
vquery = vquery & " where codigo=" & Me.txtfields(0) & """"
vquery = vquery & " and codigo_tipo_actividad=" & Mid(cmbfields(1).Text, 1, InStr(1,
cmbfields(1).Text, " ") - 1) & """"
vquery = vquery & " and codigo_actividad=" & Mid(cmbfields(2).Text, 1, InStr(1,
cmbfields(2).Text, " ") - 1) & """"
rs.Open vquery, db, adOpenStatic, adLockOptimistic

```



```

If rs.fields(0) = 0 Then
    MsgBox "El Grupo que quiere actualizar NO EXISTE", vbCritical, "Mensaje del Sistema EDEN"
    Exit Sub
End If
rs.Close
vquery = "update grupo-turistico set nombre=" & Trim(txtfields(1)) & ","
vquery = vquery & "propietario=" & Trim(txtfields(2)) & ","
vquery = vquery & "representante=" & Trim(txtfields(3)) & ","
vquery = vquery & "identificacion=" & Trim(txtfields(13)) & ","
vquery = vquery & "razon_social=" & Trim(txtfields(4)) & ","
vquery = vquery & "codigo_categoria=" & Trim(scategoria) & ","
vquery = vquery & "tipo-local=" & Trim(stipolocal) & ","
vquery = vquery & "codigoqprovincia=" & Trim(sprovincia) & ","
vquery = vquery & "codigo_canton=" & Trim(scanton) & ","
vquery = vquery & "codigo_sector=" & Trim(ssector) & ","
vquery = vquery & "codigoarroquia=" & Trim(sparroquia) & ","
vquery = vquery & "ubicacion=" & Trim(txtfields(11)) & ","
vquery = vquery & "telefono=" & Trim(txtfields(8)) & ","
vquery = vquery & "email=" & Trim(txtfields(10)) & ","
vquery = vquery & "fax=" & Trim(txtfields(9)) & ","
vquery = vquery & "fecha-operacion=" & Trim(txtfields(7)) & ","
vquery = vquery & "fecha-ingreso=" & Trim(txtfields(6)) & ","
vquery = vquery & "Observacion=" & Trim(txtfields(14)) & ","
If Mid(cmbfields(0).Text, 1, InStr(1, cmbfields(0).Text, " ") - 1) = "I" Then
    vquery = vquery & "fecha-clausura=" & Trim(txtfields(12)) & ","
End If
vquery = vquery & "folio=" & Trim(txtfields(5)) & ","
vquery = vquery & "estado=" & Trim(estado) & " "
vquery = vquery & "where codigo=" & Trim(txtfields(0)) & " and "
vquery = vquery & "codigo_tipo_actividad=" & Trim(scodigotipo) & " and "
vquery = vquery & "codigo_actividad=" & Trim(sactividad) & ""
End If
'Manda a ejecutar el query
db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
For Each oText In Me.txtfields
    oText.Text = Empty
Next
For Each oText In Me.cmbfields
    oText.Text = Empty
Next
For Each oText In Me.mebfields
    oText.Mask = ""
    oText.Text = ""
    oText.Mask = "##-??-####"
Next
Call bloquear(False)
SetButtons "WDATE"
Exit Sub
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Sub
Private Sub Form_Load()
    Call bloquear(False)
    Me.Width = 9105

```

```

Me.Height = 6660
cmdclausura.Visible = False
End Sub
Private Sub mebfields_LostFocus(Index As Integer)
Select Case Index
Case 0
txtfields(6).Text = mebfields(0).Text
'mebfields(1).SetFocus
Case 1
txtfields(7).Text = mebfields(1).Text
'mebfields(2).SetFocus
Case 2
txtfields(12).Text = mebfields(2).Text
End Select
End Sub

```

frmgrupoinstalacion

```

Public codgrupo As String
Public codtipoactiv As String
Public codactiv As String
Public codtipoinst As String
Public codinst As String
Public coddetalle As String
Public carga As Boolean
Public cargains As Boolean
Public Sub borrarinst(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As String, cdinst As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
query = "delete from grupo-instalacion where codigo_grupo=" & cdgrupo & """"
query = query & " and codigo_tipo_actividad=" & cdtipoactiv & """"
query = query & " and codigo_actividad=" & cdactiv & """"
query = query & " and codigo_instalacion=" & cdinst & """"
query = query & " and codigo_tipo_instalacion=" & cdtipoinst & """"
db.Execute query
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Public Function valor_det(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As String, cdinst As String, cddetalle As String) As String
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "select cantidad from grupo-instalacion"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & """"
vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & """"
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & """"
vquery = vquery & " and codigo_instalacion=" & Trim(cdinst) & """"

```

```

vquery = vquery & " and codigo_tipo_instalacion=" & Trim(cdtipoint) & ""
vquery = vquery & " and codigo_detalle_instalacion=" & cddetalle & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.EOF Then
    valor-det = Empty
Else
    If Not IsNull(rs("cantidad")) Then
        valor-det = rs("cantidad")
    Else
        valor-det = Empty
    End If
End If
rs.Close
db.Close
Exit Function
AddErr:
    MsgBox Err.Description
End Function
Public Function es_seleccionado_det(cdgrupo As String, cdtipoactiv As String, cdactiv As String,
cdtipoint As String, cdinst As String, cddetalle As String) As Byte
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta-base
    vquery = "select cantidad from grupo-instalacion"
    vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & ""
    vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
    vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
    vquery = vquery & " and codigo_instalacion=" & Trim(cdinst) & ""
    vquery = vquery & " and codigo_tipo_instalacion=" & Trim(cdtipoint) & ""
    vquery = vquery & " and codigo_detalle_instalacion=" & cddetalle & ""
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    If rs.EOF Then
        es-seleccionado-det=0
    Else
        es-seleccionadodet = 1
    End If
rs.Close
db.Close
Exit Function
AddErr:
    MsgBox Err.Description
End Function
Public Function es_seleccionado(cdgrupo As String, cdtipoactiv As String, cdactiv As String,
cdtipoint As String, cdinst As String) As Byte
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta-base
    vquery = "select codigo_instalacion from grupo-instalacion"
    vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & ""
    vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""

```

```

vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
vquery = vquery & " and codigo_instalacion=" & Trim(cdinst) & ""
vquery = vquery & " and codigo_tipo_instalacion=" & Trim(cdtipoint) & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.EOF Then
    es-seleccionado = 0
Else
    es-seleccionado = 1
End If
rs.Close
db.Close
Exit Function
AddErr:
    MsgBox Err.Description
End Function
Public Sub actualizar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As
String, cdinst As String, cddetalle As String, valor As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    query = "update grupo-instalacion set cantidad=" & valor & " where codigo_grupo=" & cdgrupo
& ""
    query = query & " and codigo_tipo_actividad=" & cdtipoactiv & ""
    query = query & " and codigo_actividad=" & cdactiv & ""
    query = query & " and codigo_instalacion=" & cdinst & ""
    query = query & " and codigo_tipo_instalacion=" & cdtipoint & ""
    query = query & " and codigo_detalle_instalacion=" & cddetalle & ""
    db.Execute query
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Public Sub borrar_temporal(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    query = "delete from grupo-instalacion-t where codigo_grupo=" & cdgrupo & ""
    query = query & " and codigo_tipo_actividad=" & cdtipoactiv & ""
    query = query & " and codigo_actividad=" & cdactiv & ""
    db.Execute query
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Public Sub llenar_maestro(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    vquery = "insert into grupo-instalacion select * from grupo-instalacion-t"

```

```

    vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & "' and
codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
    vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
    db.Execute vquery
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Public Sub llenar_temporal(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    vquery = "insert into grupo_instalacion select * from grupo_instalacion"
    vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & "' and
codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
    vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
    db.Execute vquery
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Public Sub insertar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As
String, cdinst As String, cddetalle As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    vquery = "insert into
grupo_instalacion(codigo_grupo,codigo_tipo_actividad,codigo_actividad,codigo_instalacion,codigo
_tipo_instalacion,codigo_detalle_instalacion,cantidad)"
    vquery = vquery & " VALUES ("
    vquery = vquery & "" & Trim(cdgrupo) & "," & Trim(cdtipoactiv) & "," & Trim(cdactiv) &
"," & Trim(cdinst) & "," & Trim(cdtipoint) & "," & Trim(cddetalle) & ",1)"
    db.Execute vquery
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Public Sub borrar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As String,
cdinst As String, cddetalle As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    query = "delete from grupo_instalacion where codigo_grupo=" & cdgrupo & ""
    query = query & " and codigo_tipo_actividad=" & cdtipoactiv & ""
    query = query & " and codigo_actividad=" & cdactiv & ""
    query = query & " and codigo_instalacion=" & cdinst & ""
    query = query & " and codigo_tipo_instalacion=" & cdtipoint & ""
    query = query & " and codigo_detalle_instalacion=" & cddetalle & ""

```



```

    db.Execute query
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub llenardetalle(ByRef vspread As vaSpread)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta_base
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    cnt = 1
    vspread.MaxRows = cnt
    codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
    codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
    codgrupo = Me.txtcodigo
    codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
    vaSpread1.Col = 1
    codinst = vaSpread1.Text
    carga = True
    While Not rs.EOF
        vspread.Row = cnt
        vspread.Col = 1
        vspread.Text = rs.fields(0)
        vspread.Col = 3
        vspread.Text = rs.fields(1)
        vspread.Col = 2
        vspread.Value = es_seleccionado_det(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst,
rs.fields(0))
        If vspread.Value = 1 Then
            vspread.Col = 4
            vspread.Value = valor_det(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst, rs.fields(0))
            vspread.Lock = False
        Else
            vspread.Col = 4
            vspread.Value = Empty
            vspread.Lock = True
        End If
        cnt = cnt + 1
        vspread.MaxRows = cnt
        rs.MoveNext
    Wend
    carga = False
    rs.Close
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub llenarinstalacion(ByRef vspread As vaSpread)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection

```

```

Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
cnt = 1
vspread.MaxRows = 0
vspread.MaxRows = cnt
codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
cargains = True
While Not rs.EOF
    vspread.Row = cnt
    vspread.Col = 1
    vspread.Text = rs.fields(0)
    vspread.Col = 2
    vspread.Value = es_seleccionado(codgrupo, codtipoactiv, codactiv, codtipoinst, rs.fields(0))
    vspread.Col = 3
    vspread.Text = rs.fields(1)
    vspread.Col = 4
    vspread.Lock = True
    cnt = cnt + 1
    vspread.MaxRows = cnt
    rs.MoveNext
Wend
cargains = False
rs.Close
db.Close
vspread.Row = vspread.ActiveRow
vspread.Col = vspread.ActiveCol
vspread.Text = vspread.Text
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cbotipo_Click()
Dim codtipo As String
If Trim(cbotipo.Text) <> "" Then
    codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
    vaSpread2.MaxRows = 0
    vaSpread2.MaxRows = 1
    vquery = "select codigo,descripcion from instalacion where codigo_tipo_instalacion=" &
codtipoinst & """"
    Call llenarinstalacion(vaSpread1)
End If
End Sub
Private Sub cmdcerrar_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
'1.-Borro el maestro
db.Open ruta-base
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
query = "delete from grupo-instalacion where codigo_grupo=" & codgrupo & """"
query = query & " and codigo_tipo_actividad=" & codtipoactiv & """"
query = query & " and codigo_actividad=" & codactiv & """"

```

```

db.Execute query
db.Close
'2.-traspaso lo del temporal al maestro
Call llenar_maestro(codgrupo, codtipoactiv, codactiv)
'3.-borro el temporal
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub cmdguardar_Click()
'borro el temporal y deajo como está
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub Form_Activate()
'Llena codigos
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
Call llenar_temporal(codgrupo, codtipoactiv, codactiv)
Me.cbotipo.Clear
vquery = "select codigo,descripcion from tipo-instalacion"
Call llenacombo(cbotipo, 1)
End Sub
Private Sub vaSpread1_ButtonClicked(ByVal Col As Long, ByVal Row As Long, ByVal
ButtonDown As Integer)
vaSpread1.Row = Row
vaSpread1.Col = Col
If vaSpread1.Col = 2 Then
If vaSpread1.Value = "1" And cargains = False Then
vquery = "select codigo,descripcion from detalle-instalacion where estado='A'"
Call llenardetalle(vaSpread2)
Else
Me.vaSpread2.MaxRows = 0
Me.vaSpread2.MaxRows = 1
End If
End If
End Sub
Private Sub vaSpread1_Click(ByVal Col As Long, ByVal Row As Long)
vaSpread1.Row = Row
vaSpread1.Col = 2
If vaSpread1.Value = "1" Then
vquery = "select codigo,descripcion from detalle-instalacion where estado='A'"
Call llenardetalle(vaSpread2)
Else
Me.vaSpread2.MaxRows = 0
Me.vaSpread2.MaxRows = 1
End If
End Sub
Private Sub vaSpread1_KeyDown(KeyCode As Integer, Shift As Integer)
vaSpread1.EditMode = False
End Sub
Private Sub vaSpread2_ButtonClicked(ByVal Col As Long, ByVal Row As Long, ByVal
ButtonDown As Integer)
vaSpread2.Row = Row
vaSpread2.Col = Col
If vaSpread2.MaxRows > 0 Then
vaSpread2.Row = Row
vaSpread2.Col = Col
codtipoint = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)

```

```

codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
vaSpread2.Col = 1
coddetalle = vaSpread2.Text
vaSpread1.Col = 1
codinst = vaSpread1.Text
vaSpread2.Col = Col
If carga = False Then
    If vaSpread2.Value = "1" Then
        Call insertar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst, coddetalle)
        vaSpread2.Col = 4
        vaSpread2.Text = "1"
        vaSpread2.Lock = False
    Else
        Call borrar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst, coddetalle)
        vaSpread2.Col = 4
        vaSpread2.Lock = True
        vaSpread2.Value = Empty
    End If
End If
End Sub
vaSpread2.Col = Col
End Sub
Private Sub vaSpread2_EditMode(ByVal Col As Long, ByVal Row As Long, ByVal Mode As Integer, ByVal ChangeMade As Boolean)
Dim valor As String
Select Case Mode
Case 0
    vaSpread2.Row = Row
    vaSpread2.Col = 4
    valor = vaSpread2.Text
    If Trim(valor) <> Empty Then
        codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
        codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
        codgrupo = Me.txtcodigo
        codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
        vaSpread2.Col = 1
        coddetalle = vaSpread2.Text
        vaSpread1.Col = 1
        codinst = vaSpread1.Text
        Call actualizar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst, coddetalle, valor)
    End If
End Select
End Sub

```

frmgrupopago

```

Public codgrupo As String
Public codtipoactiv As String
Public codactiv As String
Public codtipoinst As String
Public codinst As String
Public coddetalle As String
Public carga As Boolean
Public cargains As Boolean
Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean

```

```

Dim mbDataChanged As Boolean
Dim fotoruta As String
Dim valor As Double
Public Sub marcargrid()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim i As Long
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
valor = 0
For i = 1 To vaSpread2.MaxRows - 1
    vaSpread2.Row = i
    vaSpread2.Col = 1
    vquery = "select * from cartera-detalle where codigo=" & Trim(txtsecuencial.Text)
    vquery = vquery & " and codigo_tipo_pago=" & Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") -
1) & ""
    vquery = vquery & " and codigo_motivo_pago=" & Trim(vaSpread2.Value) & ""
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    If Not rs.EOF Then
        vaSpread2.Col = 2
        vaSpread2.Value = 1
        vaSpread2.Col = 4
        If Not IsNull(rs.fields(5)) Then
            vaSpread2.Row = i
            vaSpread2.Value = rs.fields(5)
        End If
    End If
    rs.Close
Next i
txtvalor.Text = suma
db.Close
End Sub
Public Sub cargar_forma()
Dim i As Integer
If Adodc2.Recordset.EOF Or Adodc2.Recordset.BOF Then
    If Adodc2.Recordset.EOF Then
        MsgBox "No se recuperaron registros", vbInformation
        Exit Sub
    End If
    Adodc2.Recordset.MoveLast
End If
For i = 0 To cbogrupos.ListCount - 1
    If Trim(Adodc2.Recordset.fields(2)) = Mid(cbogrupos.List(i), 1, InStr(1, cbogrupos.List(i), " ") - 1)
Then
        codgrupo = Mid(cbogrupos.List(i), InStr(1, cbogrupos.List(i), " ") + 1)
        If Trim(Adodc2.Recordset.fields(3)) = Mid(codgrupo, 1, InStr(1, codgrupo, " ") - 1) Then
            codactiv = Mid(codgrupo, InStr(1, codgrupo, " ") + 1)
            If Trim(Adodc2.Recordset.fields(4)) = Mid(codactiv, 1, InStr(1, codactiv, " ") - 1) Then
                cbogrupos.ListIndex = i
            End If
        End If
    End If
Next i
codtipoint = Trim(Adodc2.Recordset.fields(1))
For i = 0 To cbotipo.ListCount - 1
    If codtipoint = Mid(cbotipo.List(i), 1, InStr(1, cbotipo.List(i), " ") - 1) Then
        cbotipo.ListIndex = i
    End If

```



```

Next i
If Not IsNull(Adodc2.Recordset.fields(5)) Then
    txtrecibo.Text = Adodc2.Recordset.fields(5)
End If
If Not IsNull(Adodc2.Recordset.fields(6)) Then
    mebfields.Text = Format(Adodc2.Recordset.fields(6), "dd-mmm-yyyy")
End If
If Not IsNull(Adodc2.Recordset.fields(0)) Then
    txtsecuencial.Text = Adodc2.Recordset.fields(0)
End If
Call marcargrid
End Sub
Public Function suma() As Long
Dim i As Long
valor = 0
For i = 1 To vaSpread2.MaxRows - 1
    vaSpread2.Col = 2
    vaSpread2.Row = i
    If vaSpread2.Value = "1" Then
        vaSpread2.Col = 4
        valor = valor + Val(vaSpread2.Value)
    End If
Next i
suma = valor
End Function
Public Function secuencial() As Long
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not rs.EOF Then
    secuencial = rs.fields(0) + 1
Else
    secuencial = 1
End If
rs.Close
db.Close
End Function
Private Sub SetButtons(accion As String)
Select Case accion
Case "ADD"
    cmdadd.Visible = False
    cmdconsultar.Visible = False
    cmdcerrar.Visible = False
    cmdcancel.Visible = True
    cmdupdate.Visible = True
    cmdeliminar.Visible = False
    cmdejecutar.Visible = False
Case "QUERY"
    cmdadd.Visible = False
    cmdconsultar.Visible = False
    cmdcerrar.Visible = False
    cmdcancel.Visible = True
    cmdupdate.Visible = False
    cmdeliminar.Visible = False
    cmdejecutar.Visible = True
Case "EXECUTE"

```

```

cmdadd.Visible = False
cmdconsultar.Visible = False
cmdcerrar.Visible = False
cmdcancel.Visible = True
cmdupdate.Visible = False
cmdeliminar.Visible = True
cmdejecutar.Visible = False
Case "DEL"
cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
Case "CANCEL"
cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
Case "UPDATE"
cmdadd.Visible = True
cmdconsultar.Visible = True
cmdcerrar.Visible = True
cmdcancel.Visible = False
cmdupdate.Visible = False
cmdeliminar.Visible = False
cmdejecutar.Visible = False
End Select
End Sub
Private Sub bloquear(valor As Boolean)
txtrecibo.Enabled = valor
cbogruppo.Enabled = valor
cbotipo.Enabled = valor
mebfields.Enabled = valor
txtvalor.Enabled = False
vaSpread2.Enabled = valor
txtsecuencial.Enabled = valor
End Sub
Public Sub llenacombo1(ByRef combo As ComboBox, icodigos As Integer)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If icodigos = 1 Then
While Not rs.EOF
combo.AddItem rs.fields(0) & " " & rs.fields(1) & " " & rs.fields(2) & " " & rs.fields(3)
rs.MoveNext
Wend
Else
While Not rs.EOF
combo.AddItem rs.fields(0) & " " & rs.fields(1)
rs.MoveNext
Wend

```

```

End If
rs.Close
db.Close
End Sub
'ok
Public Sub borrarinst(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As
String, cdinst As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "delete from grupo-servicio where codigo_grupo=" & cdgrupo & """"
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & """"
vquery = vquery & " and codigo_actividad=" & cdactiv & """"
vquery = vquery & " and codigo_servicio=" & cdinst & """"
vquery = vquery & " and codigo_tipo_servicio=" & cdtipoint & """"
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok
Public Function valor_det(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint
As String, cdinst As String) As String
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta_base
vquery = "select cantidad from grupo-servicio"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & """"
vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & """"
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & """"
vquery = vquery & " and codigo_servicio=" & Trim(cdinst) & """"
vquery = vquery & " and codigo_tipo_servicio=" & Trim(cdtipoint) & """"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.EOF Then
valor-det = Empty
Else
If Not IsNull(rs("cantidad")) Then
valor-det = rs("cantidad")
Else
valor-det = Empty
End If
End If
rs.Close
db.Close
Exit Function
AddErr:
MsgBox Err.Description
End Function
Public Function es_seleccionado(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdinst
As String, cdtipoint As String) As Byte
Dim rs As ADODB.Recordset

```



```

Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta-base
    vquery = "select codigo-servicio from grupo_servicio"
    vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & ""
    vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
    vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
    vquery = vquery & " and codigo-servicio=" & Trim(cdtipoinst) & ""
    vquery = vquery & " and codigo_tipo_servicio=" & Trim(cdinst) & ""
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    If rs.EOF Then
        es-seleccionado = 0
    Else
        es-seleccionado = 1
    End If
    rs.Close
    db.Close
Exit Function
AddErr:
    MsgBox Err.Description
End Function
'ok
Public Sub actualizar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As
String, cdinst As String, valor As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta-base
    vquery = "update grupo-servicio set cantidad=" & valor & " where codigo_grupo=" & cdgrupo
& ""
    vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & ""
    vquery = vquery & " and codigo_actividad=" & cdactiv & ""
    vquery = vquery & " and codigo_servicio=" & cdinst & ""
    vquery = vquery & " and codigo_tipo_servicio=" & cdtipoinst & ""
    db.Execute vquery
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
'ok
Public Sub borrar_temporal(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open ruta_base
    vquery = "delete from grupo-serviciot where codigo_grupo=" & cdgrupo & ""
    vquery = vquery & " and codigo-tipo-actividad=" & cdtipoactiv & ""
    vquery = vquery & " and codigo_actividad=" & cdactiv & ""
    db.Execute vquery
    db.Close
Exit Sub

```


'ok

```
Public Sub borrar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As String, cdinst As String)
```

```
Dim rs As ADODB.Recordset
```

```
Dim db As ADODB.Connection
```

```
On Error GoTo AddErr
```

```
Set db = New ADODB.Connection
```

```
db.CursorLocation = adUseClient
```

```
db.Open ruta-base
```

```
vquery = "delete from grupo-servicio where codigo_grupo=" & cdgrupo & ""
```

```
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & ""
```

```
vquery = vquery & " and codigo_actividad=" & cdactiv & ""
```

```
vquery = vquery & " and codigo_servicio=" & cdinst & ""
```

```
vquery = vquery & " and codigo_tipo_servicio=" & cdtipoinst & ""
```

```
db.Execute vquery
```

```
db.Close
```

```
Exit Sub
```

```
AddErr:
```

```
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub llenarinstalacion(ByRef vspread As vaSpread)
```

```
Dim rs As ADODB.Recordset
```

```
Dim db As ADODB.Connection
```

```
Dim cnt As Long
```

```
On Error GoTo AddErr
```

```
Set db = New ADODB.Connection
```

```
Set rs = New ADODB.Recordset
```

```
db.CursorLocation = adUseClient
```

```
db.Open ruta_base
```

```
rs.Open vquery, db, adOpenStatic, adLockOptimistic
```

```
cnt = 1
```

```
vspread.MaxRows = 0
```

```
vspread.MaxRows = cnt
```

```
cargains = True
```

```
While Not rs.EOF
```

```
vspread.Row = cnt
```

```
vspread.Col = 1
```

```
vspread.Text = rs.fields(0)
```

```
vspread.Col = 2
```

```
vspread.Col = 3
```

```
vspread.Text = rs.fields(1)
```

```
vspread.Col = 2
```

```
If vspread.Value = 1 Then
```

```
vspread.Col = 4
```

```
vspread.Value = valor_det(codgrupo, cdtipoactiv, cdactiv, cdtipoinst, cdinst)
```

```
vspread.Lock = False
```

```
Else
```

```
vspread.Col = 4
```

```
vspread.Value = Empty
```

```
vspread.Lock = True
```

```
End If
```

```
cnt = cnt + 1
```

```
vspread.MaxRows = cnt
```

```
rs.MoveNext
```

```
Wend
```

```
cargains = False
```

```
rs.Close
```

```
db.Close
```

```
vspread.Row = vspread.ActiveRow
```

```
vspread.Col = vspread.ActiveCol
```

```

Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub Adodc2_MoveComplete(ByVal adReason As ADODB.EventReasonEnum, ByVal
pError As ADODB.Error, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As
ADODB.Recordset)
Call cargar_forma
End Sub
Private Sub cbotipo_Click()
Dim codtipo As String
If Trim(cbotipo.Text) <> "" Then
    codtipoint = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
    vaSpread2.MaxRows = 0
    vaSpread2.MaxRows = 1
    vquery = "select codigo,descripcion from motivo_pago where codigo_tipo_pago=" & codtipoint
    & ""
    vquery = vquery & " order by descripcion"
    Call llenarinstalacion(vaSpread2)
End If
If mbAddNewFlag = True Then
    If Trim(cbotipo.Text) <> "" Then
        vquery = "select max(codigo) from cartera where codigo_tipo_pago=" & Mid(cbotipo.Text, 1,
InStr(cbotipo.Text, " ") - 1) & ""
        txtsecuencial.Text = secuencial
        txtsecuencial.Enabled = False
    End If
End If
End Sub
Private Sub cmdCancel_Click()
Dim i As Long
i = 0
mebfields.Mask = ""
mebfields.Text = ""
mebfields.Mask = "##-??-####"
txtrecibo = ""
txtsecuencial = ""
txtvalor = ""
'llena estado
mbAddNewFlag = False
cbogrupos.ListIndex = -1
cbotipo.ListIndex = -1
vaSpread2.MaxRows = 0
vaSpread2.MaxRows = 1
Call bloquear(False)
valor = "0"
SetButtons "CANCEL"
End Sub
Private Sub cmdcerrar_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
'1.-Borro el maestro
db.Open ruta_base
vquery = "delete from grupo-servicio where codigo_grupo=" & codgrupo & ""
vquery = vquery & " and codigo_tipo_actividad=" & codtipoactiv & ""
vquery = vquery & " and codigo_actividad=" & codactiv & ""
db.Execute vquery

```

```

db.Close
'2.-traspaso lo del temporal al maestro
Call llenar_maestro(codgrupo, codtipoactiv, codactiv)
'3.-borro el temporal
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub cmdguardar_Click()
'borro el temporal y dejo como está
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
On Error GoTo AddErr
i = 0
mebfields.Mask = "##-??-####"
'llena estado
cbogrupos.ListIndex = -1
cbotipos.ListIndex = -1
txtrecibo = ""
txtsecuencial = ""
txtvalor = ""
vaSpread2.MaxRows = 0
vaSpread2.MaxRows = 0
Call bloquear(True)
mbAddNewFlag = True
SetButtons "ADD"
cbogrupos.SetFocus
valor = "0"
Exit Sub
AddErr:
MsgBox Err.Description
Resume Next
End Sub
Private Sub cmdconsultar_Click()
Dim i As Long
On Error GoTo AddErr
i = 0
mebfields.Mask = "##-??-####"
'llena estado
cbogrupos.ListIndex = -1
cbotipos.ListIndex = -1
txtrecibo = ""
txtsecuencial = ""
txtvalor = ""
vaSpread2.MaxRows = 0
vaSpread2.MaxRows = 0
Call bloquear(True)
mbAddNewFlag = False
SetButtons "QUERY"
cbogrupos.SetFocus
valor = "0"
Exit Sub
AddErr:
MsgBox Err.Description
Resume Next
End Sub
Private Sub cmdejecutar_Click()
Dim rs As ADODB.Recordset

```

```

Dim db As ADODB.Connection
Dim vtemp As String
If IsDate(mebfields.Text) = False Then
    mebfields.Mask = ""
    mebfields.Text = ""
End If
vquery = "select * from cartera "
If Trim(cbogrupo.Text) <> "" Then
    codgrupo = Mid(cbogrupo.Text, 1, InStr(1, cbogrupo.Text, " ") - 1)
    tmp = Mid(cbogrupo.Text, InStr(1, cbogrupo.Text, " ") + 1)
    codtipoactiv = Mid(tmp, 1, InStr(1, tmp, " ") - 1)
    tmp = Mid(tmp, InStr(1, tmp, " ") + 1)
    codactiv = Mid(tmp, 1, InStr(1, tmp, " ") - 1)
    vquery = vquery & " where codigo_grupo=" & codgrupo & """"
    vquery = vquery & " and codigo_tipo_actividad=" & Trim(codtipoactiv) & """"
    vquery = vquery & " and codigo_actividad=" & Trim(codactiv) & """"
    If Trim(cbotipo.Text) <> "" Then
        codtipoint = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
        vquery = vquery & " and codigo_tipo_pago=" & Trim(codtipoint) & """"
    End If
    If Trim(txtsecuencial.Text) <> "" Then
        vquery = vquery & " and codigo=" & Trim(txtsecuencial.Text)
    End If
    If Trim(txtrecibo.Text) <> "" Then
        vquery = vquery & " and recibo=" & Trim(txtrecibo.Text)
    End If
    If IsDate(mebfields.Text) Then
        vquery = vquery & " and fecha=" & Trim(mebfields.Text) & """"
    End If
ElseIf Trim(cbotipo.Text) <> "" Then
    codtipoint = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
    vquery = vquery & " where codigo_tipo_pago=" & Trim(codtipoint) & """"
    If Trim(txtsecuencial.Text) <> "" Then
        vquery = vquery & " and codigo=" & Trim(txtsecuencial.Text)
    End If
    If Trim(txtrecibo.Text) <> "" Then
        vquery = vquery & " and recibo=" & Trim(txtrecibo.Text)
    End If
    If Trim(mebfields.Text) <> "" Then
        vquery = vquery & " and fecha=" & Trim(mebfields.Text) & """"
    End If
ElseIf Trim(txtsecuencial.Text) <> "" Then
    vquery = vquery & " where codigo=" & Trim(txtsecuencial.Text)
    If Trim(txtrecibo.Text) <> "" Then
        vquery = vquery & " and recibo=" & Trim(txtrecibo.Text)
    End If
    If Trim(mebfields.Text) <> "" Then
        vquery = vquery & " and fecha=" & Trim(mebfields.Text) & """"
    End If
ElseIf Trim(txtrecibo.Text) <> "" Then
    vquery = vquery & " where recibo=" & Trim(txtrecibo.Text)
    If Trim(mebfields.Text) <> "" Then
        vquery = vquery & " and fecha=" & Trim(mebfields.Text) & """"
    End If
ElseIf Trim(mebfields.Text) <> "" Then
    vquery = vquery & " where fecha=" & Trim(mebfields.Text) & """"
End If
mebfields.Mask = "##-??-####"
Adodc2.RecordSource = vquery
Adodc2.Refresh

```

```

SetButtons "EXECUTE"
End Sub
Private Sub cmdeliminar_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim vmsgbox As Integer
Dim j As Integer
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
vmsgbox = MsgBox("Esta seguro de ELIMINAR este registro", vbOKCancel, "Mensaje del
Sistema EDEN")
If (vmsgbox = vbCancel) Then
Exit Sub
End If
db.Open ruta-base
'Empieza transacción
db.BeginTrans
vquery = "delete from cartera where codigo=" & txtsecuencial.Text & " and codigo_tipo_pago="
& Trim(codtipoinst) & ""
'Manda a ejecutar el query
db.Execute vquery
'Termina la transacción
db.CommitTrans
db.Close
Call bloquear(False)
SetButtons "DEL"
MsgBox "El registro se ELIMINO exitosamente", vbInformation, "Mensaje Sistema EDEN"
Unload Me
Exit Sub
AddErr:
MsgBox Err.Description
db.RollbackTrans
Resume Next
End Sub
Private Sub cmdUpdate_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim tmp As String
Dim motivo As String
Dim i As Integer
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
'valido fecha
If IsDate(mebfields.Text) = False Then
mebfields.Mask = ""
mebfields.Text = ""
mebfields.Mask = "##-??-####"
End If
'valido campos requeridos
If (cbogrupos.Text = "" Or cbotipos.Text = "" Or txtrecibo.Text = "" Or txtvalor.Text = "" Or
mebfields.Text = "") Then
MsgBox "Ingrese todos los campos requeridos", vbExclamation, "Mensaje del Sistema EDEN"
Exit Sub
End If
codgrupo = Mid(cbogrupos.Text, 1, InStr(1, cbogrupos.Text, " ") - 1)

```

```

tmp = Mid(cbogruppo.Text, InStr(1, cbogruppo.Text, " ") + 1)
codtipoactiv = Mid(tmp, 1, InStr(1, tmp, " ") - 1)
tmp = Mid(tmp, InStr(1, tmp, " ") + 1)
codactiv = Mid(tmp, 1, InStr(1, tmp, " ") - 1)
codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
'busco duplicados
vquery = "select count(*) from cartera "
vquery = vquery & "where codigo=" & txtsecuencial.Text & ""
vquery = vquery & " and codigo_tipo_pago=" & Trim(codtipoinst) & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.fields(0) > 0 Then
    MsgBox "El Registro de Pago ya Existe", vbCritical, "Mensaje del Sistema EDEN"
    Exit Sub
End If
rs.Close
vquery = "insert into
cartera(codigo,codigo_tipo_pago,codigo_grupo,codigo_tipo_actividad,codigo_actividad,recibo,fecha
a,fecha_sistema ) VALUES"
vquery = vquery & "(" & txtsecuencial.Text & "," & codtipoinst & ""
vquery = vquery & "," & codgmpo & ""
vquery = vquery & "," & codtipoactiv & ""
vquery = vquery & "," & codactiv & ""
vquery = vquery & "," & txtrecibo.Text & ""
vquery = vquery & "," & mebfields.Text & ""
vquery = vquery & "," & Format(Date, "dd-mmm-yyyy") & ""
db.Execute vquery
For i = 1 To Me.vaSpread2.MaxRows - 1
    vaSpread2.Col = 2
    vaSpread2.Row = i
    If vaSpread2.Value = "1" Then
        vaSpread2.Col = 4
        tmp = vaSpread2.Text
        vaSpread2.Col = 1
        motivo = vaSpread2.Text
        vquery = "insert into cartera_detalle(codigo,codigo_tipo_pago,codigo_motivo_pago,valor)"
        vquery = vquery & "VALUES(" & txtsecuencial.Text & "," & codtipoinst & ""
        vquery = vquery & "," & motivo & ""
        vquery = vquery & "," & tmp & ""
        db.Execute vquery
    End If
Next i
db.Close
mebfields.Mask = ""
mebfields.Text = ""
mebfields.Mask = "##-??-####"
txtrecibo = ""
txtsecuencial = ""
txtvalor = ""
'llena estado
mbAddNewFlag = False
cbogruppo.ListIndex = - 1
cbotipo.ListIndex = - 1
vaSpread2.MaxRows = 0
vaSpread2.MaxRows = 1
Call bloquear(False)
valor = "0"
SetButtons "UPDATE"
Exit Sub
AddErr:
MsgBox Err.Description

```



```

Resume Next
End Sub
Private Sub Form_Activate()
    Call bloquear(False)
    vquery = "select codigo,codigo_tipo_actividad,codigo_actividad,nombre from grupo-turistico
order by nombre"
    Call llenacombo1(cbogrupos, 1)
    vquery = "select codigo,descripcion from tipo_pago where estado='A' order by descripcion"
    Call llenacombo(cbotipos, 1)
    Adodc2.ConnectionString = ruta-base
End Sub
Private Sub Form_Load()
    frmgrupopago.Height = 5325
    frmgrupopago.Width = 8550
End Sub
Private Sub vaSpread2_ButtonClicked(ByVal Col As Long, ByVal Row As Long, ByVal
ButtonDown As Integer)
Dim codigo As String
vaSpread2.Row = Row
vaSpread2.Col = Col
If vaSpread2.MaxRows > 0 Then
    vaSpread2.Row = Row
    vaSpread2.Col = Col
    vaSpread2.Col = 1
    vaSpread2.Col = Col
    If cargains = False Then
        If vaSpread2.Value = "1" Then
            vaSpread2.Col = 4
            vaSpread2.Text = "1"
            vaSpread2.Lock = False
        Else
            vaSpread2.Col = 4
            vaSpread2.Lock = True
            vaSpread2.Value = Empty
        End If
    End If
End If
txtvalor.Text = suma
vaSpread2.Col = Col
End Sub
Private Sub vaSpread2_EditMode(ByVal Col As Long, ByVal Row As Long, ByVal Mode As
Integer, ByVal ChangeMade As Boolean)

Select Case Mode
Case 0
    vaSpread2.Row = Row
    vaSpread2.Col = 4
    txtvalor.Text = suma
Case 1
End Select
End Sub

```

frmgruposervicio

```

Public codgrupo As String
Public codtipoactiv As String
Public codactiv As String
Public codtipoint As String
Public codinst As String
Public coddetalle As String

```

```

Public carga As Boolean
Public cargains As Boolean
'ok
Public Sub borrarinst(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As
String, cdinst As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "delete from grupo_servicio where codigo_grupo=" & cdgrupo & """"
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & """"
vquery = vquery & " and codigo_actividad=" & cdactiv & """"
vquery = vquery & " and codigo_servicio=" & cdinst & """"
vquery = vquery & " and codigo_tipo_servicio=" & cdtipoint & """"
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok
Public Function valor_det(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint
As String, cdinst As String) As String
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "select cantidad from grupo_servicio"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & """"
vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & """"
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & """"
vquery = vquery & " and codigo_servicio=" & Trim(cdinst) & """"
vquery = vquery & " and codigo_tipo_servicio=" & Trim(cdtipoint) & """"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.EOF Then
valor-det = Empty
Else
If Not IsNull(rs("cantidad")) Then
valor-det = rs("cantidad")
Else
valor-det = Empty
End If
End If
rs.Close
db.Close
Exit Function
AddErr:
MsgBox Err.Description
End Function
'ok
Public Function es_seleccionado(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdinst
As String, cdtipoint As String) As Byte
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection

```

```

Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "select codigo_servicio from grupo_servicio"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & ""
vquery = vquery & " and codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
vquery = vquery & " and codigo_servicio=" & Trim(cdtipoint) & ""
vquery = vquery & " and codigo_tipo_servicio=" & Trim(cdinst) & ""
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If rs.EOF Then
    es-seleccionado = 0
Else
    es-seleccionado = 1
End If
rs.Close
db.Close
Exit Function
AddErr:
MsgBox Err.Description
End Function
'ok
Public Sub actualizar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoint As
String, cdinst As String, valor As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "update grupo-servicio set cantidad=" & valor & " where codigo_grupo=" & cdgrupo
& ""
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & ""
vquery = vquery & " and codigo_actividad=" & cdactiv & ""
vquery = vquery & " and codigo_servicio=" & cdinst & ""
vquery = vquery & " and codigo_tipo_servicio=" & cdtipoint & ""
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok
Public Sub borrar_temporal(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "delete from grupo-servicio-t where codigo_grupo=" & cdgrupo & ""
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & ""
vquery = vquery & " and codigo_actividad=" & cdactiv & ""
db.Execute vquery
db.Close
Exit Sub
AddErr:

```

```

MsgBox Err.Description
End Sub
'ok
Public Sub llenar_maestro(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "insert into grupo-servicio select * from grupo-serviciot"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & " and
codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok
Public Sub llenar_temporal(cdgrupo As String, cdtipoactiv As String, cdactiv As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta_base
vquery = "insert into grupo-serviciot select * from grupo_servicio"
vquery = vquery & " where codigo_grupo=" & Trim(cdgrupo) & " and
codigo_tipo_actividad=" & Trim(cdtipoactiv) & ""
vquery = vquery & " and codigo_actividad=" & Trim(cdactiv) & ""
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok
Public Sub insertar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As
String, cdinst As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "insert into
grupo_servicio(codigo_grupo,codigo_tipo_actividad,codigo_actividad,codigo_servicio,codigo_tipo
_servicio,cantidad)"
vquery = vquery & " VALUES ("
vquery = vquery & """" & Trim(cdgrupo) & """," & Trim(cdtipoactiv) & """," & Trim(cdactiv) &
"""," & Trim(cdinst) & """," & Trim(cdtipoinst) & """,1)"
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
'ok

```

```

Public Sub borrar(cdgrupo As String, cdtipoactiv As String, cdactiv As String, cdtipoinst As String,
cdinst As String)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
On Error GoTo AddErr
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "delete from grupo_servicio where codigo_grupo=" & cdgrupo & ""
vquery = vquery & " and codigo_tipo_actividad=" & cdtipoactiv & ""
vquery = vquery & " and codigo_actividad=" & cdactiv & ""
vquery = vquery & " and codigo_servicio=" & cdinst & ""
vquery = vquery & " and codigo_tipo_servicio=" & cdtipoinst & ""
db.Execute vquery
db.Close
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub llenarinstalacion(ByRef vspread As vaSpread)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
cnt = 1
vspread.MaxRows = 0
vspread.MaxRows = cnt
codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
cargains = True
While Not rs.EOF
vspread.Row = cnt
vspread.Col = 1
vspread.Text = rs.fields(0)
vspread.Col = 2
vspread.Value = es_seleccionado(codgrupo, codtipoactiv, codactiv, codtipoinst, rs.fields(0))
vspread.Col = 3
vspread.Text = rs.fields(1)
vspread.Col = 2
If vspread.Value = 1 Then
vspread.Col = 4
vspread.Value = valor_det(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst)
vspread.Lock = False
Else
vspread.Col = 4
vspread.Value = Empty
vspread.Lock = True
End If
cnt = cnt + 1
vspread.MaxRows = cnt
rs.MoveNext
Wend
cargains = False

```

```

rs.Close
db.Close
vspread.Row = vspread.ActiveRow
vspread.Col = vspread.ActiveCol
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cbotipo_Click()
Dim codtipo As String
If Trim(cbotipo.Text) <> "" Then
codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
vaSpread2.MaxRows = 0
vaSpread2.MaxRows = 1
vquery = "select codigo,descripcion from servicio where codigo_tipo_servicio=" & codtipoinst
& ""
vquery = vquery & " order by descripcion"
Call llenarinstalacion(vaSpread2)
End If
End Sub
Private Sub cmdcerrar_Click()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
'1.-Borro el maestro
db.Open ruta-base
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
vquery = "delete fi-om grupo-servicio where codigo_grupo=" & codgrupo & ""
vquery = vquery & " and codigo_tipo_actividad=" & codtipoactiv & ""
vquery = vquery & " and codigo_actividad=" & codactiv & ""
db.Execute vquery
db.Close
'2.-traspaso lo del temporal al maestro
Call llenar_maestro(codgrupo, codtipoactiv, codactiv)
'3.-borro el temporal
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub cmdguardar_Click()
'borro el temporal y dejo como está
Call borrar_temporal(codgrupo, codtipoactiv, codactiv)
Unload Me
End Sub
Private Sub Form_Activate()
'Llena codigos
codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
codgrupo = Me.txtcodigo
codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
Call llenar_temporal(codgrupo, codtipoactiv, codactiv)
Me.cbotipo.Clear
vquery = "select codigo,descripcion fi-om tipo-servicio order by descripcion"
Call llenacombo(cbotipo, 1)
End Sub
Private Sub vaSpread2_ButtonClicked(ByVal Col As Long, ByVal Row As Long, ByVal
ButtonDown As Integer)
vaSpread2.Row = Row
vaSpread2.Col = Col

```

```

If vaSpread2.MaxRows > 0 Then
    vaSpread2.Row = Row
    vaSpread2.Col = Col
    codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
    codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
    codgrupo = Me.txtcodigo
    codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
vaSpread2.Col = 1
    codinst = vaSpread2.Text
    vaSpread2.Col = Col
    If cargains = False Then
        If vaSpread2.Value = "1" Then
            Call insertar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst)
            vaSpread2.Col = 4
            vaSpread2.Text = "1"
            vaSpread2.Lock = False
        Else
            Call borrar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst)
            vaSpread2.Col = 4
            vaSpread2.Lock = True
            vaSpread2.Value = Empty
        End If
    End If
End If
vaSpread2.Col = Col
End Sub
Private Sub vaSpread2_EditMode(ByVal Col As Long, ByVal Row As Long, ByVal Mode As
Integer, ByVal ChangeMade As Boolean)
Dim valor As String
Select Case Mode
Case 0
    vaSpread2.Row = Row
    vaSpread2.Col = 4
    valor = vaSpread2.Text
    If Trim(valor) <> Empty Then
        codtipoinst = Mid(cbotipo.Text, 1, InStr(1, cbotipo.Text, " ") - 1)
        codactiv = Mid(txtactividad, 1, InStr(1, txtactividad, " ") - 1)
        codgrupo = Me.txtcodigo
        codtipoactiv = Mid(txttipoactiv, 1, InStr(1, txttipoactiv, " ") - 1)
        vaSpread2.Col = 1
        coddetalle = vaSpread2.Text
    Call actualizar(codgrupo, codtipoactiv, codactiv, codtipoinst, codinst, valor)
    End If
End Select
End Sub

```

frminstalacion

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.2
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.2
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0.4
    DataGrid1.Columns(3).Width = DataGrid1.Width * 0.2

```

```

DataGrid1.Columns(0).Caption = "COD TIPO"
DataGrid1.Columns(1).Caption = "CODIGO"
DataGrid1.Columns(2).Caption = "DESCRIPCION"
DataGrid1.Columns(3).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i=0
For Each oText In Me.txtFields
    oText.Enabled = valor
    i = i + 1
Next
End Sub
Private Sub cargartexto()
    Dim i As Long
    Dim a(1) As String
    i=0
    For Each oText In Me.txtFields
        oText.Text = PrimaryCLS.fields(i)
        Adocanton.ConnectionString = ruta_base
        If Not oText.Text = "" Then
            Select Case i
                Case 0
                    vquery = "select descripcion from tipo--instalacion where codigo=" & Trim(oText.Text) &
,,,,,
                    Adocanton.RecordSource = vquery
                    Adocanton.Refresh
                    txtDescripcion(i).Text = Adocanton.Recordset.fields(0)
                End Select
            Else
                txtDescripcion(0).Text = ""
            End If
            i = i + 1
        Next
    End Sub
Private Sub cmdbuscarprov_Click()
vquery = "select codigo,descripcion from tipo--instalacion order by descripcion"
frmbuscar.Col = 0
frmbuscar.Show vbModal
Me.txtFields(0) = vcodigo
Me.txtDescripcion(0) = vdescripcion
End Sub
Private Sub Form_Load()
    vquery = "select codigo_tipo_instalacion,codigo,descripcion,estado from instalacion Order by
descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
    cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340

```



```

End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
        End Select
    End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmdbuscarprov.Visible = True
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub

```

```

Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr
lblStatus.Caption = "Modificar registro"
cmdbuscarprov.Visible = True
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no estén vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from instalacion where codigo_tipo_instalacion=" & Me.txtFields(0) &
""
vquery = vquery & " and codigo=" & Me.txtFields(1) & ""
If duplicado And mbAddNewFlag = True Then
txtFields(0).SetFocus
MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
PrimaryCLS.fields(i) = oText.Text
i = i + 1

```

```

Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmdbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
PrimaryCLS.MoveFirst
mbDataChanged = False
Exit Sub
GoFirstError:
MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
On Error GoTo GoLastError
PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:
MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
On Error GoTo GoNextError
PrimaryCLS.MoveNext
Exit Sub
GoNextError:
MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields__KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 2 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0

```

```
End If
End If
End If
End Sub
```

frmlogin

```
Public Function conectar() As Boolean
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
vquery = "select * from usuario where usuario='" & Trim(Me.txtUserName) & "'"
vquery = vquery & " and password='" & Trim(Me.txtPassword) & "'"
rs.Open vquery, db, adOpenStatic, adLockOptimistic
If Not rs.EOF Then
conectar = True
Else
conectar = False
End If
rs.Close
Exit Function
AddErr:
MsgBox Err.Description
db.RollbackTrans
Resume Next
End Function
Private Sub cmdCancel_Click()
'establecer la variable global a false
'para indicar un inicio de sesión fallido
LoginSucceeded = False
Me.Hide
End Sub
Private Sub cmdOK_Click()
'comprobar si la contraseña es correcta
If conectar Then
'colocar código aquí para pasar al sub
'que llama si la contraseña es correcta
'lo más fácil es establecer una variable global
LoginSucceeded = True
Me.Hide
Else
MsgBox "La contraseña no es válida. Vuelva a intentarlo", , "Inicio de sesión"
txtPassword.SetFocus
SendKeys "{Home}+{End}"
intentos = intentos + 1
If intentos = 3 Then
LoginSucceeded = False
Me.Hide
End If
End Sub
Private Sub Form_Load()
intentos = 0
End Sub
```

Frmlogin1

```
Option Explicit
Public LoginSucceeded As Boolean
Public intentos As Byte
Public Sub actualizar_passwod()
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta_base
    db.BeginTrans
    vquery = "update usuario set password='" & Me.Text2 & "' where usuario='" &
Trim(Me.txtUserName) & "'"
    vquery = vquery & " and password='" & Trim(Me.txtPassword) & "'"
    db.Execute vquery
    db.CommitTrans
    db.Close
Exit Sub
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Sub
Public Function conectar() As Boolean
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Dim cnt As Long
On Error GoTo AddErr
    Set db = New ADODB.Connection
    Set rs = New ADODB.Recordset
    db.CursorLocation = adUseClient
    db.Open ruta_base
    vquery = "select * from usuario where usuario='" & Trim(Me.txtUserName) & "'"
    vquery = vquery & " and password='" & Trim(Me.txtPassword) & "'"
    rs.Open vquery, db, adOpenStatic, adLockOptimistic
    If Not rs.EOF Then
        conectar = True
    Else
        conectar = False
    End If
    rs.Close
Exit Function
AddErr:
    MsgBox Err.Description
    db.RollbackTrans
    Resume Next
End Function
Private Sub cmdCancel_Click()
    'establecer la variable global a false
    'para indicar un inicio de sesión fallido
    Unload Me
End Sub
Private Sub cmdOK_Click()
    'comprobar si la contraseña es correcta
    If conectar Then
        'colocar código aquí para pasar al sub
```

```

'que llama si la contraseña es correcta
'lo más fácil es establecer una variable global
If Me.Text1 = Me.Text2 Then
    Call actualizar_passwod
    Unload Me
Else
    MsgBox "La confirmación es errónea. Vuelva a intentarlo", , "Cambio de Contraseña"
    Me.Text2.SetFocus
    SendKeys "{Home}+ {End}"
End If
Else
    MsgBox "La contraseña no es válida. Vuelva a intentarlo", , "Cambio de Contraseña"
txtPassword.SetFocus
SendKeys "{Home}+{End}"
intentos = intentos + 1
If intentos = 3 Then
    Unload Me
End If
End If
End Sub
Private Sub Form_Load()
Me.txtUserName = user
End Sub

```

frmmotivopago

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.22
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0.5
    DataGrid1.Columns(3).Width = DataGrid1.Width * 0
    DataGrid1.Columns(0).Caption = "COD TIPO"
    DataGrid1.Columns(1).Caption = "CODIGO"
    DataGrid1.Columns(2).Caption = "DESCRIPCION"
    DataGrid1.Columns(3).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
    oText.Enabled = valor
    i = i + 1
Next
For Each oText In Me.Optestado
    oText.Enabled = valor
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
    oText.Text = PrimaryCLS.fields(i)

```

```

Adocanton.ConnectionString = ruta_base
If Not oText.Text = "" Then
    Select Case i
        Case 0
            vquery = "select descripcion from tipo_pago where codigo = " & Trim(oText.Text) & ""
            Adocanton.RecordSource = vquery
            Adocanton.Refresh
            txtDescripcion(i).Text = Adocanton.Recordset.fields(0)
        Case 3
            If txtFields(i).Text = "A" Then
                Optestado(0).Value = True
            Else
                Optestado(1).Value = True
            End If
        End Select
    End If
    i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
    vquery = "select codigo,descripcion from tipo_pago ORDER BY SIGNO,DESCRIPCION"
    frmbuscar.Col = 0
    frmbuscar.Show vbModal
    Me.txtFields(0) = vcodigo
    Me.txtDescripcion(0) = vdescripcion
End Sub
Private Sub Form_Load()
    vquery = "select codigo_tipo_pago,codigo,descripcion,estado from motivoqago Order by
descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGridView1.DataMember = "Primary"
    Set DataGridView1.DataSource = PrimaryCLS
    Me.Width = 7800
    Me.Height = 6015
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
    cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else

```

```

    cmdPrevious_Click
End If
Case vbKeyDown, vbKeyPageDown
    If Shift = vbCtrlMask Then
        cmdLast_Click
    Else
        cmdNext_Click
    End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro: " & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmdbuscarprov.Visible = True
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    cmdbuscarprov.Visible = True

```



```

mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
cmbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'Asignar el valor por default del radio button al texto
If Optestado(i).Value = True Then
txtFields(3).Text = "A"
Else
txtFields(3).Text = "I"
End If
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from motivo_pago where codigo_tipo_pago=" & Me.txtFields(0) & ""
vquery = vquery & " and codigo=" & Me.txtFields(1) & ""
If duplicado And mbAddNewFlag = True Then
txtFields(0).SetFocus
MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
PrimaryCLS.fields(i) = oText.Text
i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub

```

```

Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
    Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
    Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdRefresh.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub

```

Frmparroquial

```

Private WithEvents PrimaryCLS As clscanton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private indice As Integer
Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.15
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.15
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0.15

```

```

DataGrid1.Columns(3).Width = DataGrid1.Width * 0.5
DataGrid1.Columns(0).Caption = "COD PROV"
DataGrid1.Columns(1).Caption = "COD CANTON"
DataGrid1.Columns(2).Caption = "COD PARROQUIA"
DataGrid1.Columns(3).Caption = "DESCRIPCION"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
    oText.Enabled = valor
    i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(2) As String
i = 0
For Each oText In Me.txtFields
    oText.Text = PrimaryCLS.fields(i)
    Adoparroquia.ConnectionString = ruta_base
    If Not oText.Text = "" Then
        Select Case i
            Case 0
                vquery = "select descripcion from provincia where codigo_provincia =" & Trim(oText.Text)
                Adoparroquia.RecordSource = vquery
                Adoparroquia.Refresh
                txtDescripcion(i).Text = Adoparroquia.Recordset.fields(0)
            Case 1
                vquery = "select descripcion from canton where codigo_provincia =" &
Trim(PrimaryCLS.fields(i - 1)) & " and codigo_canton=" & Trim(oText.Text)
                Adoparroquia.RecordSource = vquery
                Adoparroquia.Refresh
                txtDescripcion(i).Text = Adoparroquia.Recordset.fields(0)
        End Select
    End If
    i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
Select Case indice
Case 0
    vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
Case 1
    If Trim(txtFields(0)) <> "" Then
        vquery = "select codigo_canton,descripcion from canton where codigo_provincia=" &
Trim(Me.txtFields(0))
        vquery = vquery & " order by descripcion"
    Else
        MsgBox "Primero escoja provincia", vbInformation
        txtFields(0).SetFocus
    End Sub
End If
End Select
frmbuscar.Col = 0
frmbuscar.Show vbModal
Me.txtFields(indice) = vcodigo
Me.txtDescripcion(indice) = vdescripcion
End Sub
Private Sub Form_Load()

```

```

vquery = "select codigo_provincia,codigo_canton,codigo_parroquia,descripcion from parroquia
Order by descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub
Select Case KeyCode
Case vbKeyEscape
cmdClose_Click
Case vbKeyEnd
cmdLast_Click
Case vbKeyHome
cmdFirst_Click
Case vbKeyUp, vbKeyPageUp
If Shift = vbCtrlMask Then
cmdFirst_Click
Else
cmdPrevious_Click
End If
Case vbKeyDown, vbKeyPageDown
If Shift = vbCtrlMask Then
cmdLast_Click
Else
cmdNext_Click
End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
'Esto mostrará la posición de registro actual para este Recordset
lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
Call cargartexto
End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
On Error GoTo AddErr
i=0
For Each oText In Me.txtFields
oText.Text = Empty
i = i + 1
Next
'Muestro buscar provincias

```

```

Me.cmdbuscarprov.Visible = True
Call bloquear(True)
Me.txtFields(0).SetFocus
lblStatus.Caption = "Agregar registro"
mbAddNewFlag = True
SetButtons False
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
PrimaryCLS.Delete
Call FormatoGrid
Call cargartexto
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr

lblStatus.Caption = "Modificar registro"
cmdbuscarprov.Visible = True
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no estén vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical

```

```

Exit Sub
End If
Next
vquery = "select count(*) from parroquia where codigo_provincia=" & Me.txtFields(0)
vquery = vquery & " and codigo_canton=" & Me.txtFields(1)
vquery = vquery & " and codigo_parroquia=" & Me.txtFields(2)
If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmdbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
Exit Sub

```

GoPrevError:

```
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_GotFocus(Index As Integer)
indice = Index
If Index < 2 Then
cmdbuscarprov.Top = txtFields(Index).Top
End If
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 3 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0
End If
End If
End If
End Sub
```

frmprovincia

```
Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private indice As Integer
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.18
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.18
DataGrid1.Columns(2).Width = DataGrid1.Width * 0.18
DataGrid1.Columns(3).Width = DataGrid1.Width * 0.4
DataGrid1.Columns(0).Caption = "COD PAIS"
DataGrid1.Columns(1).Caption = "COD REGION"
DataGrid1.Columns(2).Caption = "COD PROV"
DataGrid1.Columns(3).Caption = "DESCRIPCION"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For i = 0 To 3
txtFields(i).Enabled = valor
Next
End Sub
Private Sub cargartexto()
Dim i As Long
```

```

Dim a(2) As String
i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)
Adoprov.ConnectionString = ruta_base
If Not oText.Text = "" Then
Select Case i
Case 0
vquery = "select descripcion from pais where codigo=" & Trim(oText.Text) & ""
Adoprov.RecordSource = vquery
Adoprov.Refresh
txtdescripcion(i).Text = Adoprov.Recordset.fields(0)
Case 1
vquery = "select descripcion from region where codigo=" & Trim(oText.Text) & ""
Adoprov.RecordSource = vquery
Adoprov.Refresh
txtdescripcion(i).Text = Adoprov.Recordset.fields(0)
End Select
End If
i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
Select Case indice
Case 0
vquery = "select codigo,descripcion from pais order by descripcion"
Case 1
If Trim(txtFields(0)) <> "" Then
vquery = "select codigo,descripcion from region where codigo_pais=" & Trim(Me.txtFields(0))
& ""
vquery = vquery & " order by descripcion"
Else
MsgBox "Primero escoja país", vbInformation
txtFields(0).SetFocus
Exit Sub
End If
End Select
frmbuscar.Col = 0
frmbuscar.Show vbModal
Me.txtFields(indice) = vcodigo
Me.txtdescripcion(indice) = vdescripcion
End Sub
Private Sub Form_Load()
vquery = "select codigo_pais,codigo_region,codigo_provincia,descripcion from provincia Order
by descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340

```



```

End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
        End Select
    End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro: " & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmdbuscarprov.Visible = True
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()

```

```

'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
  On Error GoTo EditErr
  lblStatus.Caption = "Modificar registro"
  cmdbuscarprov.Visible = True
  mbEditFlag = True
  Call bloquear(True)
  SetButtons False
  Exit Sub
EditErr:
  MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
  On Error Resume Next
  mbAddNewFlag = False
  PrimaryCLS.Cancel
  PrimaryCLS.Requery
  Call FormatoGrid
  SetButtons True
  Call bloquear(False)
  cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
  On Error GoTo UpdateErr
  'HACER TODAS LAS VERIFICACIONES
  'Verificando que los campos no esten vacíos
  For Each oText In Me.txtFields
    If Trim(oText.Text) = Empty Then
      oText.SetFocus
      MsgBox "Campo vacío no se puede grabar", vbCritical
      Exit Sub
    End If
  Next
  vquery = "select count(*) from provincia where codigo_provincia=" & Me.txtFields(2)
  If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
    Exit Sub
  End If
  'FIN DE VERIFICACIONES
  'graba registro nuevo
  If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
  End If
  i = 0
  For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
  Next
  mbAddNewFlag = False
  PrimaryCLS.Update

```

```

Call bloquear(False)
Me.cmdbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
PrimaryCLS.MoveFirst
mbDataChanged = False
Exit Sub
GoFirstError:
MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
On Error GoTo GoLastError
PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:
MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
On Error GoTo GoNextError
PrimaryCLS.MoveNext
Exit Sub
GoNextError:
MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_GotFocus(Index As Integer)
indice = Index
If Index < 2 Then
cmdbuscarprov.Top = txtFields(Index).Top
End If
End Sub
Private Sub txtFields_KeyUp(Index As Integer, KeyCode As Integer, Shift As Integer)
If txtanterior <> "" Then

```

```

txtFields(Index) = txtanterior
End If
End Sub

```

frmreportes

```

Public Sub llenacombo(ByRef combo As ComboBox, icodigos As Integer)
Dim rs As ADODB.Recordset
Dim db As ADODB.Connection
Set db = New ADODB.Connection
Set rs = New ADODB.Recordset
db.CursorLocation = adUseClient
db.Open ruta-base
rs.Open vquery, db, adOpenStatic, adLockOptimistic
While Not rs.EOF
    combo.AddItem rs.fields(0) & " " & rs.fields(1)
    rs.MoveNext
Wend
rs.Close
db.Close
End Sub
Private Sub Cmbfields_Click(Index As Integer)
    Select Case Index
        Case 0
            If (Cmbfields(0).Text <> "") Then
                Cmbfields(1).Clear
                vquery = "select codigo,descripcion from actividad"
                vquery = vquery & " Where codigo_tipo_actividad=" & Mid(Cmbfields(0).Text, 1, InStr(1,
Cmbfields(0).Text, " ") - 1) & """"
                vquery = vquery & " order by descripcion"
                Call llenacombo(Cmbfields(1), 2)
            End If
        Case 3
            If Trim(Cmbfields(3).Text) <> "" Then
                Cmbfields(4).Clear
                Cmbfields(5).Clear
                Cmbfields(6).Clear
                vquery = "select codigo_canton,descripcion from canton"
                vquery = vquery & " Where codigo_provincia=" & Mid(Cmbfields(3).Text, 1, InStr(1,
Cmbfields(3).Text, " ") - 1)
                vquery = vquery & " order by descripcion"
                Call llenacombo(Me.Cmbfields(4), 2)
            End If
        Case 4
            If Trim(Cmbfields(3).Text) <> "" And Trim(Cmbfields(4).Text) <> "" Then
                Cmbfields(5).Clear
                Cmbfields(6).Clear
                vquery = "select codigo_parroquia,descripcion from parroquia "
                vquery = vquery & " Where codigo_canton=" & Mid(Cmbfields(4).Text, 1, InStr(1,
Cmbfields(4).Text, " ") - 1)
                vquery = vquery & " and codigo_provincia=" & Mid(Cmbfields(3).Text, 1, InStr(1,
Cmbfields(3).Text, " ") - 1)
                vquery = vquery & " order by descripcion"
                Call llenacombo(Me.Cmbfields(5), 2)
            End If
        Case 5
            If Trim(Cmbfields(4).Text) <> "" And Trim(Cmbfields(3).Text) <> "" And
Trim(Cmbfields(5).Text) <> "" Then
                Cmbfields(6).Clear
                vquery = "select codigo,descripcion from sector"

```

```

    vquery = vquery & " Where codigo_canton=" & Mid(Cmbfields(4).Text, 1, InStr(1,
Cmbfields(4).Text, " ") - 1)
    vquery = vquery & " and codigo_provincia=" & Mid(Cmbfields(3).Text, 1, InStr(1,
Cmbfields(3).Text, " ") - 1)
    vquery = vquery & " and codigo_parroquia=" & Mid(Cmbfields(5).Text, 1, InStr(1,
Cmbfields(5).Text, " ") - 1)
    vquery = vquery & " order by descripcion"
    Call llenacombo(Me.Cmbfields(6), 2)
End If
End Select
End Sub
Private Sub Cmdimprimir_Click()
Dim query As String
cryreport.ReportFileName = rutarpt & "\grupo_turistico_param.rpt"
cryreport.PrintFileName = "grupo_turistico_param.rpt"
'limpio la variable
vquery = ""
'tipo actividad
If Cmbfields(0).Text <> "" Then
    vquery = "grupo_turistico.codigo_tipo_actividad=" & Mid(Cmbfields(0).Text, 1, InStr(1,
Cmbfields(0).Text, " ") - 1) & ""
End If
'actividad
If Cmbfields(1).Text <> "" Then
    If vquery <> "" Then
        vquery = vquery & " and grupo_turistico.codigo_actividad=" & Mid(Cmbfields(1).Text, 1,
InStr(1, Cmbfields(1).Text, " ") - 1) & ""
    Else
        vquery = vquery & "grupo_turistico.codigo_actividad=" & Mid(Cmbfields(1).Text, 1, InStr(1,
Cmbfields(1).Text, " ") - 1) & ""
    End If
End If
'categoria
If Cmbfields(2).Text <> "" Then
    If vquery <> "" Then
        vquery = vquery & " and grupo_turistico.codigo_categoria=" & Mid(Cmbfields(2).Text, 1,
InStr(1, Cmbfields(2).Text, " ") - 1) & ""
    Else
        vquery = vquery & "grupo_turistico.codigo_categoria=" & Mid(Cmbfields(2).Text, 1, InStr(1,
Cmbfields(2).Text, " ") - 1) & ""
    End If
End If
'provincia
If Cmbfields(3).Text <> "" Then
    If vquery <> "" Then
        vquery = vquery & " and grupo_turistico.codigo_provincia=" & Mid(Cmbfields(3).Text, 1,
InStr(1, Cmbfields(3).Text, " ") - 1)
    Else
        vquery = vquery & "grupo_turistico.codigo_provincia=" & Mid(Cmbfields(3).Text, 1, InStr(1,
Cmbfields(3).Text, " ") - 1)
    End If
End If
'canton
If Cmbfields(4).Text <> "" Then
    If vquery <> "" Then
        vquery = vquery & " and grupo_turistico.codigo_canton=" & Mid(Cmbfields(4).Text, 1,
InStr(1, Cmbfields(4).Text, " ") - 1)
    Else
        vquery = vquery & "grupo_turistico.codigo_canton=" & Mid(Cmbfields(4).Text, 1, InStr(1,
Cmbfields(4).Text, " ") - 1)
    End If
End If

```

```

End If
End If
'parroquia
If Cmbfields(5).Text <> "" Then
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.codigo_parroquia=" & Mid(Cmbfields(5).Text, 1,
InStr(1, Cmbfields(5).Text, " ") - 1)
  Else
    vquery = vquery & "grupo_turistico.codigo_parroquia=" & Mid(Cmbfields(5).Text, 1, InStr(1,
Cmbfields(5).Text, " ") - 1)
  End If
End If
'sector
If Cmbfields(6).Text <> "" Then
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.codigo_sector=" & Mid(Cmbfields(6).Text, 1, InStr(1,
Cmbfields(6).Text, " ") - 1)
  Else
    vquery = vquery & "grupo_turistico.codigo_sector=" & Mid(Cmbfields(6).Text, 1, InStr(1,
Cmbfields(6).Text, " ") - 1)
  End If
End If
'estado
If Cmbfields(7).Text <> "" Then
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.estado=" & Mid(Cmbfields(7).Text, 1, InStr(1,
Cmbfields(7).Text, " ") - 1) & ""
  Else
    vquery = vquery & "grupo_turistico.estado=" & Mid(Cmbfields(7).Text, 1, InStr(1,
Cmbfields(7).Text, " ") - 1) & ""
  End If
End If
'tipo local
If Cmbfields(8).Text <> "" Then
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.tipo_local=" & Mid(Cmbfields(8).Text, 1, InStr(1,
Cmbfields(8).Text, " ") - 1) & ""
  Else
    vquery = vquery & "grupo_turistico.tipo_local=" & Mid(Cmbfields(8).Text, 1, InStr(1,
Cmbfields(8).Text, " ") - 1) & ""
  End If
End If
'fecha ingreso
If Mskfields(0).Text <> " _ / / _ " And Mskfields(1).Text <> " _ / / _ " Then
  vquery = ""
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.fecha_ingreso >= #" & Mskfields(0).Text & "#" and
grupo_turistico.fecha_ingreso <= #" & Mskfields(1).Text & "#"
  Else
    vquery = vquery & "grupo_turistico.fecha_ingreso >= #" & CDate(Mskfields(0).Text) & "#" and
grupo_turistico.fecha_ingreso <= #" & CDate(Mskfields(1).Text) & "#"
    vquery = vquery & "{grupo_turistico.fecha_ingreso} >= #" & CDate("01-sep-2000") & "" and
{grupo_turistico.fecha_ingreso} <= #" & CDate("30-sep-2000") & ""
  End If
End If
'fecha clausura
If Mskfields(2).Text <> " _ / / _ " And Mskfields(3).Text <> " _ / / _ " Then
  If vquery <> "" Then
    vquery = vquery & " and grupo_turistico.fecha_clausura >= #" & CDate(Mskfields(2).Text) &
"# and grupo_turistico.fecha_clausura <= #" & CDate(Mskfields(3).Text) & "#"
  End If

```

```

Else
    vquery = vquery & "grupo_turistico.fecha_clausura >= #" & Mskfields(2).Text & "# and
{grupo_turistico.fecha_clausura} <= #" & Mskfields(3).Text & "#"
    End If
End If
query = "SELECT"
query = query & " grupo_turistico.`codigo`, grupo_turistico.`nombre`,
grupo_turistico.`propietario`, grupo_turistico.`razon_social`, grupo_turistico.`ubicacion`,
grupo_turistico.`telefono`, grupo_turistico.`fax`, grupo_turistico.`fecha_ingreso`,
grupo_turistico.`fecha_clausura`, grupo_turistico.`folio`, grupo_turistico.`identificacion`, "
query = query & " actividad.`descripcion`, "
query = query & " parroquia.`descripcion`, "
query = query & " provincia.`descripcion`, "
query = query & " tipo_actividad.`descripcion`, "
query = query & " canton.`descripcion`, "
query = query & " categoria.`descripcion`"
query = query & " From"
query = query & " (((('grupo_turistico' grupo_turistico INNER JOIN 'parroquia' parroquia
ON"
query = query & " grupo_turistico.`codigo_provincia` = parroquia.`codigo_provincia` AND"
query = query & " grupo_turistico.`codigo_canton` = parroquia.`codigo_canton` AND"
query = query & " grupo_turistico.`codigo_parroquia` = parroquia.`codigo_parroquia`)"
query = query & " INNER JOIN 'provincia' provincia ON"
query = query & " grupo_turistico.`codigo_provincia` = provincia.`codigo_provincia`)"
query = query & " INNER JOIN 'tipo_actividad' tipo_actividad ON"
query = query & " grupo_turistico.`codigo_tipo_actividad` = tipo_actividad.`codigo`)"
query = query & " INNER JOIN 'canton' canton ON"
query = query & " grupo_turistico.`codigo_provincia` = canton.`codigo_provincia` AND"
query = query & " grupo_turistico.`codigo_canton` = canton.`codigo_canton`)"
query = query & " INNER JOIN 'categoria' categoria ON"
query = query & " grupo_turistico.`codigo_categoria` = categoria.`codigo`)"
query = query & " INNER JOIN 'actividad' actividad ON"
query = query & " grupo_turistico.`codigo_actividad` = actividad.`codigo` AND"
query = query & " grupo_turistico.`codigo_tipo_actividad` = actividad.`codigo_tipo_actividad`"
query = query & " WHERE " & vquery
query = query & " Order By"
query = query & " provincia.`descripcion` ASC,"
query = query & " canton.`descripcion` ASC,"
query = query & " parroquia.`descripcion` ASC,"
query = query & " tipo_actividad.`descripcion` ASC,"
query = query & " actividad.`descripcion` ASC,"
query = query & " categoria.`descripcion` ASC"
If vquery <> "" Then
    Me.cryreport.SQLQuery = query
    Me.cryreport.Action = 1
End If
End Sub
Private Sub Command2_Click()
    Unload Me
End Sub
Private Sub Form_Load()
    Frmreportes.Height = 4725
    Frmreportes.Width = 9015
    'tipo actividad
    Cmbfields(0).Clear
    vquery = "select codigo,descripcion from tipo_actividad where estado='A' order by descripcion"
    Call llenacombo(Cmbfields(0), 2)
    'categoria
    Cmbfields(2).Clear
    vquery = "select codigo,descripcion from categoria where estado='A' order by descripcion"

```

```

Call llenacombo(Cmbfields(2), 2)
'provincia
Cmbfields(3).Clear
vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
Call llenacombo(Cmbfields(3), 2)
'estado
Cmbfields(7).Clear
vquery = "select codigo,descripcion from estado-grupo where estado='A' order by descripcion"
Call llenacombo(Cmbfields(7), 2)
'tipo local
Cmbfields(8).Clear
vquery = "select codigo,descripcion from tipo-local order by descripcion"
Call llenacombo(Cmbfields(8), 2)
End Sub

```

frmsector

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private indice As Integer
Private Sub FormatoGrid()
    DataGrid1.Columns(0).Width = DataGrid1.Width * 0.15
    DataGrid1.Columns(1).Width = DataGrid1.Width * 0.15
    DataGrid1.Columns(2).Width = DataGrid1.Width * 0.2
    DataGrid1.Columns(3).Width = DataGrid1.Width * 0.5
    DataGrid1.Columns(0).Caption = "COD PROV"
    DataGrid1.Columns(1).Caption = "COD CANTON"
    DataGrid1.Columns(2).Caption = "COD PARROQUIA"
    DataGrid1.Columns(3).Caption = "COD SECTOR"
    DataGrid1.Columns(4).Caption = "DESCRIPCION"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
    'oText.DataMember = "Primary"
    'Set oText.DataSource = PrimaryCLS
    oText.Enabled = valor
    i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(3) As String
i = 0
For Each oText In Me.txtFields
    oText.Text = PrimaryCLS.fields(i)
    Adosector.ConnectionString = ruta-base
    If Not oText.Text = "" Then
        Select Case i
            Case 0
                vquery = "select descripcion from provincia where codigo_provincia =" & Trim(oText.Text)
                Adosector.RecordSource = vquery
                Adosector.Refresh
                txtDescripcion(i).Text = Adosector.Recordset.fields(0)

```



```

Case 1
    vquery = "select descripcion from canton where codigo_provincia =" &
Trim(PrimaryCLS.fields(i - 1))& " and codigo_canton=" & Trim(oText.Text)
    Adosector.RecordSource = vquery
    Adosector.Refresh
    txtDescripcion(i).Text = Adosector.Recordset.fields(0)
Case 2
    vquery = "select descripcion from parroquia where codigo_provincia =" &
Trim(PrimaryCLS.fields(i - 2)) & " and codigo_canton=" & Trim(PrimaryCLS.fields(i - 1))& "
and codigo_parroquia=" & Trim(oText.Text)
    Adosector.RecordSource = vquery
    Adosector.Refresh
    txtDescripcion(i).Text = Adosector.Recordset.fields(0)
End Select
End If
i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
    Select Case indice
    Case 0
        vquery = "select codigo_provincia,descripcion from provincia order by descripcion"
    Case 1
        If Trim(txtFields(0)) <> "" Then
            vquery = "select codigo_canton,descripcion from canton where codigo_provincia=" &
Trim(Me.txtFields(0))
            vquery = vquery & " order by descripcion"
        Else
            MsgBox "Primero escoja provincia", vbInformation
            txtFields(0).SetFocus
        End If
    Case 2
        If Trim(txtFields(0)) <> "" Then
            If Trim(txtFields(1)) <> "" Then
                vquery = "select codigo_parroquia,descripcion from parroquia where codigo_provincia=" &
Trim(Me.txtFields(0))
                vquery = vquery & " and codigo_canton=" & Trim(Me.txtFields(1))
                vquery = vquery & " order by descripcion"
            Else
                MsgBox "Primero escoja canton", vbInformation
                txtFields(0).SetFocus
            End If
        Else
            MsgBox "Primero escoja provincia", vbInformation
            txtFields(0).SetFocus
        End If
    End Select
    frmbuscar.Col = 0
    frmbuscar.Show vbModal
    Me.txtFields(indice) = vcodigo
    Me.txtDescripcion(indice) = vdescripcion
End Sub
Private Sub Form_Load()
    vquery = "select codigo_provincia,codigo_canton,codigo_parroquia,codigo,descripcion from
sector Order by descripcion"
    Set PrimaryCLS = New cls Canton
    Dim oText As TextBox

```

```

'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
    Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmdbuscarprov.Visible = True
    Call bloquear(True)
    Me.txtFields(0).SetFocus
    lblStatus.Caption = "Agregar registro"

```

```

mbAddNewFlag = True
SetButtons False
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
PrimaryCLS.Delete
Call FormatoGrid
Call cargartexto
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
'Esto sólo es necesario en aplicaciones multiusuario
On Error GoTo RefreshErr
PrimaryCLS.Requery
Call FormatoGrid
Call cargartexto
Exit Sub
RefreshErr:
MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
On Error GoTo EditErr
lblStatus.Caption = "Modificar registro"
cmbbuscarprov.Visible = True
mbEditFlag = True
Call bloquear(True)
SetButtons False
Exit Sub
EditErr:
MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
cmbbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from sector where codigo_provincia=" & Me.txtFields(0)
vquery = vquery & " and codigo_canton=" & Me.txtFields(1)

```

```

vquery = vquery & " and codigo_parroquia=" & Me.txtFields(2)
vquery = vquery & " and codigo=" & Me.txtFields(3)
If duplicado And mbAddNewFlag = True Then
    txtFields(0).SetFocus
    MsgBox "Registro duplicado", vbCritical
    Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
    PrimaryCLS.AddNew
End If
i=0
For Each oText In Me.txtFields
    PrimaryCLS.fields(i) = oText.Text
    i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmdbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError
    PrimaryCLS.MoveFirst
    mbDataChanged = False
    Exit Sub
GoFirstError:
    MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
    On Error GoTo GoLastError
    PrimaryCLS.MoveLast
    mbDataChanged = False
    Exit Sub
GoLastError:
    MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
    On Error GoTo GoNextError
    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError
    PrimaryCLS.MovePrevious
    Exit Sub
GoPrevError:
    MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)

```

```

cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_GotFocus(Index As Integer)
indice = Index
If Index < 3 Then
cmdbuscarprov.Top = txtFields(Index).Top
End If
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 4 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0
End If
End If
End If
End If
End Sub

```

frmservicio

```

Private WithEvents PrimaryCLS As clscanton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.2
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.2
DataGrid1.Columns(2).Width = DataGrid1.Width * 0.4
DataGrid1.Columns(3).Width = DataGrid1.Width * 0.2
DataGrid1.Columns(0).Caption = "CODIGO TIPO"
DataGrid1.Columns(1).Caption = "CODIGO"
DataGrid1.Columns(2).Caption = "DESCRIPCION"
DataGrid1.Columns(3).Caption = "ESTADO"
End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
oText.Enabled = valor
i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
oText.Text = PrimaryCLS.fields(i)

```

```

Adocanton.ConnectionString = ruta_base
If Not oText.Text = "" Then
    Select Case i
        Case 0
            vquery = "select descripcion from tipo-servicio where codigo =" & Trim(oText.Text) & ""
            Adocanton.RecordSource = vquery
            Adocanton.Refresh
            txtDescripcion(i).Text = Adocanton.Recordset.fields(0)
        End Select
    End If
    i = i + 1
Next
End Sub
Private Sub cmdbuscarprov_Click()
    vquery = "select codigo,descripcion from tipo-servicio order by descripcion"
    frmbuscar.Col = 0
    frmbuscar.Show vbModal
    Me.txtFields(0) = vcodigo
    Me.txtDescripcion(0) = vdescripcion
End Sub
Private Sub Form_Load()
    vquery = "select codigo_tipo_servicio,codigo,descripcion,estado from servicio Order by
descripcion"
    Set PrimaryCLS = New clscanton
    Dim oText As TextBox
    'Enlaza los cuadros de texto con el proveedor de datos
    DataGrid1.DataMember = "Primary"
    Set DataGrid1.DataSource = PrimaryCLS
    Call FormatoGrid
    Call cargartexto
    Call bloquear(False)
    cmdbuscarprov.Visible = False
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    lblStatus.Width = Me.Width - 1500
    cmdNext.Left = lblStatus.Width + 700
    cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select

```

```

End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    'Muestro buscar provincias
    Me.cmbbuscarprov.Visible = True
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
    Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    cmbbuscarprov.Visible = True
    mbEditFlag = True
    Call bloquear(True)
    SetButtons False
    Exit Sub
EditErr:
    MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()

```

```

On Error Resume Next
mbAddNewFlag = False
PrimaryCLS.Cancel
PrimaryCLS.Requery
Call FormatoGrid
SetButtons True
Call bloquear(False)
cmbbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from servicio where codigo_tipo_servicio=" & Me.txtFields(0) & ""
vquery = vquery & " and codigo=" & Me.txtFields(1) & ""
If duplicado And mbAddNewFlag = True Then
txtFields(0).SetFocus
MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
PrimaryCLS.fields(i) = oText.Text
i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
Me.cmbbuscarprov.Visible = False
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
PrimaryCLS.MoveFirst
mbDataChanged = False
Exit Sub
GoFirstError:
MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
On Error GoTo GoLastError

```



```

PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:
MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()
On Error GoTo GoNextError
PrimaryCLS.MoveNext
Exit Sub
GoNextError:
MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 2 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 12 Then
KeyAscii = 0
End If
End If
End If
End Sub

```

frmtipoactividad

```

Private WithEvents PrimaryCLS As clsconton
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
Dim txtanterior As String
Private Sub FormatoGrid()
DataGrid1.Columns(0).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(1).Width = DataGrid1.Width * 0.5
DataGrid1.Columns(2).Width = DataGrid1.Width * 0
DataGrid1.Columns(3).Width = DataGrid1.Width * 0.22
DataGrid1.Columns(0).Caption = "CODIGO"
DataGrid1.Columns(1).Caption = "DESCRIPCION"
DataGrid1.Columns(2).Caption = "COMENTARIO"
DataGrid1.Columns(3).Caption = "ESTADO"

```

```

End Sub
Private Sub bloquear(valor As Boolean)
Dim i As Long
i = 0
For Each oText In Me.txtFields
    oText.Enabled = valor
    i = i + 1
Next
End Sub
Private Sub cargartexto()
Dim i As Long
Dim a(1) As String
i = 0
For Each oText In Me.txtFields
    oText.Text = PrimaryCLS.fields(i)
    Adocanton.ConnectionString = ruta-base
    i = i + 1
Next
End Sub
Private Sub Form_Load()
vquery = "select codigo,descripcion,comentario,estado from tipo_actividad Order by descripcion"
Set PrimaryCLS = New clscanton
Dim oText As TextBox
'Enlaza los cuadros de texto con el proveedor de datos
DataGrid1.DataMember = "Primary"
Set DataGrid1.DataSource = PrimaryCLS
Call FormatoGrid
Call cargartexto
Call bloquear(False)
End Sub
Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub
Select Case KeyCode
Case vbKeyEscape
    cmdClose_Click
Case vbKeyEnd
    cmdLast_Click
Case vbKeyHome
    cmdFirst_Click
Case vbKeyUp, vbKeyPageUp
    If Shift = vbCtrlMask Then
        cmdFirst_Click
    Else
        cmdPrevious_Click
    End If
Case vbKeyDown, vbKeyPageDown
    If Shift = vbCtrlMask Then
        cmdLast_Click
    Else
        cmdNext_Click
    End If
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)

```

```

    Screen.MousePointer = vbDefault
End Sub
Private Sub PrimaryCLS_MoveComplete()
    'Esto mostrará la posición de registro actual para este Recordset
    lblStatus.Caption = "Registro:" & CStr(PrimaryCLS.AbsolutePosition)
    If Not PrimaryCLS.EOF And Not PrimaryCLS.BOF And mbAddNewFlag = False Then
        Call cargartexto
    End If
End Sub
Private Sub cmdAdd_Click()
    Dim i As Long
    On Error GoTo AddErr
    i = 0
    For Each oText In Me.txtFields
        oText.Text = Empty
        i = i + 1
    Next
    Call bloquear(True)
    lblStatus.Caption = "Agregar registro"
    mbAddNewFlag = True
    SetButtons False
    Exit Sub
AddErr:
    MsgBox Err.Description
End Sub
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    PrimaryCLS.Delete
    Call cargartexto
    Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
Private Sub cmdRefresh_Click()
    'Esto sólo es necesario en aplicaciones multiusuario
    On Error GoTo RefreshErr
    PrimaryCLS.Requery
    Call FormatoGrid
    Call cargartexto
    Exit Sub
RefreshErr:
    MsgBox Err.Description
End Sub
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    lblStatus.Caption = "Modificar registro"
    mbEditFlag = True
    Call bloquear(True)
    SetButtons False
    Exit Sub
EditErr:
    MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
    On Error Resume Next
    mbAddNewFlag = False
    PrimaryCLS.Cancel
    PrimaryCLS.Requery
    Call FormatoGrid
    SetButtons True

```

```

Call bloquear(False)
Call cargartexto
cmdbuscarprov.Visible = False
End Sub
Private Sub cmdUpdate_Click()
Dim i As Long
On Error GoTo UpdateErr
'HACER TODAS LAS VERIFICACIONES
'Verificando que los campos no esten vacíos
For Each oText In Me.txtFields
If Trim(oText.Text) = Empty Then
oText.SetFocus
MsgBox "Campo vacío no se puede grabar", vbCritical
Exit Sub
End If
Next
vquery = "select count(*) from tipo_actividad where codigo=" & Me.txtFields(0) & ""
If duplicado And mbAddNewFlag = True Then
txtFields(0).SetFocus
MsgBox "Registro duplicado", vbCritical
Exit Sub
End If
'FIN DE VERIFICACIONES
'graba registro nuevo
If mbAddNewFlag = True Then
PrimaryCLS.AddNew
End If
i = 0
For Each oText In Me.txtFields
PrimaryCLS.fields(i) = oText.Text
i = i + 1
Next
mbAddNewFlag = False
PrimaryCLS.Update
Call bloquear(False)
SetButtons True
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
PrimaryCLS.MoveFirst
mbDataChanged = False
Exit Sub
GoFirstError:
MsgBox Err.Description
End Sub
Private Sub cmdLast_Click()
On Error GoTo GoLastError
PrimaryCLS.MoveLast
mbDataChanged = False
Exit Sub
GoLastError:
MsgBox Err.Description
End Sub
Private Sub cmdNext_Click()

```

```
On Error GoTo GoNextError
PrimaryCLS.MoveNext
Exit Sub
GoNextError:
MsgBox Err.Description
End Sub
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
PrimaryCLS.MovePrevious
Exit Sub
GoPrevError:
MsgBox Err.Description
End Sub
Private Sub SetButtons(bVal As Boolean)
cmdAdd.Visible = bVal
cmdUpdate.Visible = Not bVal
cmdCancel.Visible = Not bVal
cmdDelete.Visible = bVal
cmdClose.Visible = bVal
cmdRefresh.Visible = bVal
cmdNext.Enabled = bVal
cmdFirst.Enabled = bVal
cmdLast.Enabled = bVal
cmdPrevious.Enabled = bVal
End Sub
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
If Index < 1 Then
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
If
```

Bibliografĩa

BIBLIOGRAFIA

<http://www.drdw-.coin/components/viewprof/distribute.htm>
Dr. DWG View Professional (OCX visor del mapa)

www.appgeo.com
Consultora de Sistemas de Información Geográfico y Ambiental

www.turismo.gub.uy
Ministerio de Turismo de Uruguay

www.turismocoruna.com
Turismo de La Coruña

www.uniandes.edu.co/Colombia/Turismo/
Turismo de Colombia

www.cecer.army.mil/grass
Geographic Resources Analysis Support System

www.cgdi.gc.ca/english/
Información Geográfica del Canadá en Internet

www.fgdc.gov/
Federal Geographic Data Committee

www.shef.ac.uk
The University of Sheffield

www.gisportal.com
Geographic Information System, también conocido como Great GIS Net Sites