

Indice

1. Introducción

2. Bases de datos.

3. Programación.

4. Formularios

5. Informes

1. Introducción

En este manual daremos a conocer al estudiante el lenguaje de programación Visual Fox Pro 6.0. Se pretende familiarizar al alumno con el entorno de desarrollo de aplicaciones para base de datos.

Visual Fox Pro es un gestor de base de datos, orientado a la programación de objetos.

Visual Fox Pro pertenece a la familia xbase lo que hace que su programación sea sencilla, estructurada y mas fácil de entender tanto para programadores principiantes como programadores expertos.

Nos enfocaremos en cinco áreas principales:

- Base de datos: Trata sobre el diseño, creación y manipulación de tablas libres o tablas con integridad referencial (base de datos)
- Programación: En esta parte seremos capaces de identificar y aplicar las estructuras básicas de programación y conocer aspectos sobre la programación orientada a objetos.
- Formularios: Aplicaremos conocimientos para la integración de una interfaz con el usuario y base de datos.
- Informes: Aprenderemos a diseñar las salidas de los sistemas de información, haciendo uso de las herramientas que el programa ofrece.
- SQL: En esta sección haremos uso del lenguaje SQL para manipular datos, creando así diferentes consultas o vistas.

2. Bases de datos.

Introducción

En este manual mostraremos lo básico pero también puedes consultar algún libro de Visual FoxPro pues todo será para tu desarrollo, en las bibliotecas o en las librerías. FoxPro es un lenguaje de mucha rapidez, esto lo convierte en un lenguaje de los más rápidos en el mercado, FoxPro es también uno de los lenguajes de programación que contiene mucha potencia en el manejo de las bases de datos.

Conceptos Básicos

Base de datos es un conjunto de datos almacenados en campos y registros, es decir que la base de datos esta conformada por estos dos elementos. Así, por ejemplo:

Base de datos (Campos)

(Nombre de las variables) Nombre Edad Sueldo

(Registros) José Mendoza 19 2,568.52 Omar Mendoza 20 3,759.45

Los campos son los que están en vertical y los registros llamados también datos, son los que están en horizontal.

Componentes de Interfaz

Algunas herramientas más utilizadas son:

Ventana Examinar: una vista, tipo hoja de cálculo, de una tabla.

Ventana Código: para desplegar código asociado a varios eventos en los formularios y controles. Cuando un evento se dispara el código se ejecuta.

Ventana Depuración: permite examinar variables de memoria o valores campos y establecer puntos de interrupción. La ejecución del programa se detiene cuando una variable de memoria o una expresión con un punto de interrupción cambian de valor.

Comando Opciones (Menú de Herramientas): permite controlar la configuración de docenas de características en el entorno FoxPro, incluidos todos los comandos SET, así como planillas y bibliotecas de clases.

Ventana Propiedades: permite establecer propiedades en una buena cantidad de generadores, incluidos los generadores de formularios, informes etiquetas y de las bases de datos, también proporciona acceso a propiedades, métodos y código de eventos.

Administrador de Proyectos: un diseño completamente novedoso de FoxPro para Windows, este administrador de proyectos administra todos los componentes de un proyecto en cinco grupos:

Bases de datos (con extensión .DBC), tablas libres (con extensión .DBF), vistas locales y remotas, conexiones, etc.

Generador de consultas: una recodificación completa del RQBE (Consulta Relacional Ejemplificada), esta herramienta maneja todos los aspectos de construir una consulta.

Barras de herramientas FoxPro: proporciona a los generadores aplicaciones más de una docena de barras de herramientas para colocar toda la herramienta para varias tareas justo al alcance de sus dedos. Además, puedes diseñar tus propias barras de herramientas en conjunción con formularios, para proporcionar a los usuarios el mismo tipo de acceso instantáneo a las herramientas.

Generadores

Los generadores son entornos de trabajo en los que se construyen componentes de una aplicación de FoxPro.

En la siguiente lista te mostrare algunos generadores:

Generador de clases Para construir objetos reutilizables.

Generador de Bases de Datos Para organizar los datos en tablas y documentar las relaciones entre tablas.

Generador de formularios Para diseñar las pantallas de la aplicación.

Generador de consulta Para construir conjuntos de datos utilizados en reportes y en pantallas de sólo lectura.

Generador de informes Para construir informes para la pantalla o la impresora.

Generador de menús Construye el sistema de menús que ejecuta una aplicación.

Generador de tablas Administra el formato de las tablas utilizadas en la aplicación.

Generador de cuadrículas Permite aprender cómo las configuraciones de la propiedad de cuadrícula del objeto controlan la operación de la cuadrícula.

Asistentes

Son conjunto de cuadro de diálogos que te ayudan paso a paso a crear una determinada aplicación, por ejemplo un formulario, etc.

Asistente para formularios: Construye "Pantallas instantáneas" con la estructura de las tablas basándose en clases prediseñadas, incluidos efectos especiales en las pantallas y botones de navegación ínter construidos.

Asistente para documentación: Documenta la aplicación.

Asistente para informes: Diseña informes, sencillos o complejos, utilizando un poco más que la estructura de las tablas.

Asistente para tablas: Útil para hacer tablas sencillas

Barras de Herramientas

El propósito de que hayan estas barras es para hacerte un poco más fácil el trabajo, es decir que el uso que le dará a la ventana de comandos será un poquito reducido.

Paleta de colores creo que te imaginas que es la barra de los colores en función RGB. Bueno RGB significa (Rojo, Verde y Azul)

Generador de bases de datos en esta barra se manejan el entorno de datos, iconos para: crear, agregar, y quitar una tabla así, como también modificar, vista remota o local, examinar una tabla o editar procedimientos almacenados en el contenedor de la base de datos.

Generador de formularios esta barra te permite el paso rápido de uno a otro entre varios elementos usados en el diseño de pantallas: el entorno de los datos, la ventana propiedades, la ventana código la barra Controles de formularios, la paleta de colores, la barra de herramientas.

Distribución, entre otros.

Presentación Preliminar para que una vez diseñado un informe puedas apreciarlo como te quedará y si no te gusta pues lo podrás modificar.

Estándar Este lo verás cuando inicies FoxPro, proporciona acceso al generador de formularios y al Generador de informes, a bases de datos de impresión consultan tablas, conexiones, vistas, etiquetas, programas, clases, archivos de texto y menús.

Administrador de Proyectos

Estos proyectos están integrados por el Administrador de proyectos, quien mantiene la pista de los componentes de la aplicación. Conforme se agregan componentes a un proyecto, (Estas son las carpetas o nombre de los menús del Administrador) FoxPro los colecta bajo alguno de los siguientes encabezados:

Datos: las bases de datos (y todos los elementos que pueden describir), incluidas las tablas, vistas locales y remotas, conexiones y procedimientos almacenados, así como tablas libres y consultas.

Documentos: formularios, etiquetas e informes.

Bibliotecas de clases: repositorios de objetos usados en la aplicación.

Código: los archivos con extensión .PRG que contienen código que no está asociado con un formulario, así como bibliotecas API y archivos llamados por la aplicación.

Otros: menús, archivos de texto y otros, incluyendo mapas de bits.

La barra de herramientas

En FoxPro hay muchos botones, estos botones pertenecen a la barra de herramientas de VFP.

También existen otras barras de herramientas. Se hace clic en el menú Ver, aparece una sola opción de barras de herramientas. Sólo haz clic en la barra que quieres activar y aparecerá al igual que la estándar. Algunas de estas barras ya las vimos anteriormente, ¿recuerdas?.

Estructura de un .BDF

Los datos en FoxPro se almacenan en forma de tablas, estas tablas son las bases de datos pues la extensión de estas bases de datos es .DBF aunque también hay otras que se verá más adelante. Los DBF comienzan con una breve descripción de los datos que están en la tabla.

Escritorio de Visual FoxPro

Para ingresar a Visual FoxPro tienes que presionar el botón de inicio / todos los programas / Microsoft Visual Studio 6.0 /Microsoft Visual FoxPro 6.0 (versión windows XP).

inicio / programas / Microsoft Visual Studio 6.0 /Microsoft Visual FoxPro 6.0 (versión windows estándar).

Aparecerá en tu pantalla la siguiente ventana:

Escritorio de Visual FoxPro

Configuración de Visual FoxPro 6.0

Cuando se carga FoxPro, lee un archivo llamado CONFIG.FPW en el directorio VFP.

CONFIG.FPW es un archivo de texto ASCII y contiene instrucciones que se leen al momento de iniciar y se convierten en valores predeterminados de la sesión de FoxPro.

He aquí algunos de los elementos que se fijan en la configuración del sistema:

SAFETY = OFF && no avisa antes de sobrescribir archivos

TALK= OFF && no despliega el progreso

ECHO= OFF && inicialmente desactiva el rastreo

DEBUG= ON && habilita el rastreo y la depuración

RESOURCE=FOXUSER && elige un archivo de recursos

RESOURCE=OFF && no se use hasta que se solicite

COMMAND= DO PROG && corre PROG al inicio

Comandos de configuración (SET)

Los anteriores son comandos los cuales se pueden modificar utilizando los comandos que inician con la instrucción SET desde la ventana de comandos, por ej:

SET DEFAULT TO A: && establece el dispositivo y el directorio

Por ejemplo si los archivos que se van a utilizar se encuentran en el directorio

C:\programas\transmisión\datos, escribiremos el comando SET de la siguiente forma:

SET DEFAULT TO C:\programas\transmisión\datos.

Otros comandos de configuración son:

SET CENTURY ON/OFF

SET CLOCK ON/OFF

SET DATE

SET DELETE ON/OFF

SET ECHO ON/OFF

SET SYSMENU

Creación de tablas

Diseñador de tablas

El diseñador de tablas es una ventana donde podrás diseñar la estructura de un archivo de base de datos o tabla.

Tipos de Campos

Los campos no deben de contener caracteres especiales, fuera del carácter "Guión bajo", y tampoco puede empezar con un número. Pueden ser cuales quiera de varios tipos:

Carácter: Cualquier cosa que se pueda teclear, puede ir en un campo de caracteres. Un carácter es lo mismo que un byte y un byte puede tener cualquier valor ASCII hasta 255, con unas cuántas excepciones.

Numérico: Hasta un límite de 18 dígitos, con tantas cifras decimales como quieras. Los enteros se almacenan en este formato y un nuevo tipo de dato "moneda" se usa generalmente para dinero.

Flotante: Este no lo utilizaremos en este manual pues sólo sirve por compatibilidad con dBASE V, es equivalente al formato numérico.

Fecha: Para almacenar la fecha en un formato interno numérico, pero su validación y representación se controlan con este tipo de campo.

Lógico: Éste, ya sea verdadero o falso, es representado por T o F, para verdadero o falso respectivamente.

Memo: Representado en su estructura de registros como un campo de caracteres de longitud 10. Estos campos almacenan en una segunda tabla con la extensión .FPT. Pueden ser casi de cualquier tamaño.

General: Éstos también son campos memos, excepto que usualmente almacenan otros tipos de objetos además de texto como gráficos.

Fecha-hora: incluye la hora después de la fecha, en el formato [MM/DD/AA hh:mm:ss].

Moneda: Es un campo numérico e incluye cuatro cifras decimales.

Imagen: Usado para guardar imágenes.

Carácter binario: Campos de caracteres no sujetos a traducciones de páginas de códigos.

Memo binario: Campos memo con la misma característica NOCPTRANS.

Abrir y cerrar tablas.

Para abrir un DBF en el área de trabajo 1 cuando ninguna otra tabla está abierta, teclee en la ventana de comando (ventanita pequeña que aparece generalmente al lado izquierdo):

USE nombre de la tabla

Nota: Un área de trabajo es un espacio de memoria reservado para que allí se manipulen los datos de la tabla.

Puede especificar: USE nombre de la tabla IN 3 pero recuerda para abrir o cerrar una tabla tiene que estar ya creada. Para reservar un área de trabajo puedes digitar:

SELECT 0 && 0 es el número de área de trabajo

Este comando también puede ser SELECT nombre de la tabla / alias SELECT es para seleccionar la tabla a trabajar (abrir, cerrar, modificar, etc.) USE nombre de la tabla (y se abrirá el cuadro de dialogo abrir, pues FoxPro no sabe que tabla abrir debes abrir la que necesites).

Si necesitas tener varias tablas abiertas puedes hacer lo siguiente:

SELECT A

USE tabla1

```
SELECT B
USE tabla2
SELECT C
USE tabla3 ...
```

Y cuando quiera utilizar una de las tablas solo tienes que llamar el área de trabajo como por ejemplo si vas a utilizar la tabla2 escribe en la ventana de comandos o dentro de un programa SELECT B.

Para cerrar tablas o una sesión de tablas abiertas puedes escribir los siguientes comandos:

```
CLOSE ALL && cierra todas las tablas abiertas
CLOSE DATABASE && cierra las bases de datos abiertas
O simplemente escribe USE.
```

Diseño de tablas

Primero hay que crear un directorio para almacenar tu trabajo (aunque esto no es necesario pero si vas a trabajar con un disco flexible o en un directorio distinto al que trae designado FoxPro tienes que hacerlo), para ello digita: MD ruta luego para que FoxPro trabaje con este directorio debes digitar:

```
SET DEFA TO ruta (nombre del directorio que se supone ya creaste con el comando md).
```

Bien ahora tienes varias opciones para crear la tabla:

1. digita CREATE nombre de la tabla.
2. haz clic en nuevo (new) luego selecciona tabla (table) luego digite el nombre de la tabla.
3. Ctrl + n y selecciona tabla (table) luego digite el nombre de la tabla

Después tendrás que darle un nombre a la tabla y lo guardas aparecerá el generador de tablas con las carpetas tabla e índice (si es VFP 6.0 estará también campos).

Ahora puedes introducir el nombre, el tipo y la longitud del campo en el orden correcto.

Por ejemplo:

Nombre Carácter 40

Dirección Carácter 40

Ciudad Carácter 22

Saldo monetario 8 (este es automático)

No oprimas enter porque el programa entenderá que la tabla ya esta terminada si quieres pasar a digitar el siguiente campo debe moverte con la tecla tab NO CON ENTER. Enseguida introduce datos a tu tabla. Si digitas LIST STRUCTURE (en la ventana de comandos), verás la estructura de tu tabla puedes borrar los campos si digitas DELETE NEXT numero de campo (2 por ejemplo), y puedes recuperarlo si digitas RECALL NEXT 2 y regresará.

Ahora puedes digitar BROWSE y veras los datos que introdujiste la tabla después de creada, Si digitas SET SCAPE ON podrás detener algunos procesos cuando lo desees como el comando LIST pero aquí no será necesario pues se introducirán pocos datos en la tabla. tú puedes digitar LIST y ver los datos en forma rápida la sintaxis de LIST es:

```
LIST | FIELDS listacampos | TO PRINT | WHILE expresiondominio | FOR expresiondominio | OFF
FIELD digitas el nombre del campo o campos a visualizar. TO PRINT en para enviar la salida a la
impresora. FOR seguido de una expresión de dominio, lista sólo aquellos registros que coincidan
con la expresión. ejemplo LIST FOR ciudad ="San Salvador" el WHILE seria LIST WHILE ciudad =
"San Salvador", el WHILE se utiliza para optimizar muchos aspectos del desempeño de FoxPro.
```

OFF desactiva la opción de numeración de registros digítalos y notarás la diferencia.

Sum si digitas SUM saldo te sumará todos los saldos (registros) almacenados en este campo.

Cómo moverse en una tabla de FoxPro

Cuando utilizamos inicialmente una tabla de FoxPro, tú estás ubicado en el primer registro de la tabla. Puedes mover este puntero de registro imaginario alrededor de la tabla, en diversas formas:

Skip Mueve el puntero hacia adelante un registro

Skip-1 mueve el puntero hacia atrás un registro

Go TOP Mueve el puntero al primer registró

Go bottom Mueve el puntero al último registro en la tabla

Go 2 Mueve el puntero del registro al registro número 2.

* Prueba estos comandos primero digita uno de ellos y luego digita list y verás los resultados.

Practica 1

Según la siguiente hoja, evalúa los datos que pudan servir para realizar una tabla de Inventario, construye la estructura, tomando en cuenta algunos datos que puedan enriquecer la tabla.

Computadoras SYSTEM S.A.

Inventario físico al _____ de _____ del 2001

Código	Descripción	Marca	Exist.	Precio Costo	Precio Venta	Proveedor
025	Monitor 14"	Acer	50	900.00	1080.00	Procom S.A.
026	Monitor 12"	Acer	102	850.00	1020.00	Procom S.A.
027	Monitor Radius	Acer	10	1100.00	1320.00	Procom S.A.
A01	Case Mini tower Deluxe	ADS	300	105.00	126.00	Distribuidora SalCom
A02	Case Desk Top	ADS	100	125.00	150.00	Distribuidora SalCom
IC1	Impresor Color JET 1000	Canon	25	367.00	440.40	Procom S.A.
IC2	Impresor color JET 2000	Canon	15	452.25	542.70	Procom S.A.

Agrega 20 registros a la tabla.

Utiliza los comandos Use, Browse, List y Display, según te indique el instructor y anota en tu cuaderno lo que pasa.

Manipulación de registros

Puedes utilizar diferentes tipos de comando para manipular los datos de tu tabla, estos comandos pueden utilizarse desde su modo más simple hasta su forma compuesta utilizando expresiones de dominio, por ejemplo:

LIST FOR país = "ES" && lista los registro que país = "ES"

He aquí algunos comandos con sus expresiones de dominio:

DISPLAY ALL

DISPLAY RECORD 5

GOTO 17

GO TOP

GO BOTTOM

SKIP -3

DELETE RECORD 6

LIST FOR precio >1200

BROWSE FOR precio >=900 .AND. precio <2500

GO 6

RECALL

PACK

ZAP

Mencionaremos otros comandos que son de gran ayuda al momento de manipular los datos:

APPEND

Agrega registros a la base de datos seleccionada

APPEND FROM

Agrega registros desde otra tabla al final de los registros de la tabla seleccionada

COPY FILE

Copia cualquier tipo de archivos

COPY STRUCTURE	Almacena la estructura de una base de datos en una nueva base
COPY TO	Copia datos de una base de datos a un archivo
COUNT	Determina el número de registro en la base de datos
DISPLAY FILES	Muestra información de base de datos
DISPLAY MEMORY	Despliega la información de las variables de memoria
DISPLAY STATUS	Despliega información de la condición de FOX
DISPLAY STRUCTURE	Muestra la estructura de una base de datos
MODIFY COMMAND	Abre una ventana de modificación de texto
MODIFY STRUCTURE	Modifica la estructura de una base de datos
REPLACE	Actualiza los registros en una base de datos

Otro tipo de comandos útiles para la manipulación de tablas y registros son las FUNCIONES, las cuales se dividen en:

- a. Cadenas o manejo de caracteres: este grupo de funciones gestiona el manejo de caracteres.
- b. Matemáticas: permiten efectuar operaciones con valores numéricos.
- c. Manejo de fechas y hora: se encarga de gestionar las variables de tipo fecha.
- d. Lógicas: Se encargan de efectuar operaciones de tipo lógico.
- e. Funciones de conversión de tipos: convierte variables de un tipo a otro.

Antes de mostrar ejemplos definiremos dos componentes que se hacen necesarios en la manipulación de datos:

Constante: Son valores predefinidos que no cambian nunca por ejemplo: valor1=15478, cadena="abcd".

Variable: Se utilizan como identificadores de secciones de memoria donde se guarda información, por ejemplo: var1=campo1, var2=tabla.campo.

Las variables se pueden definir como: públicas, privadas y locales ejemplo:

PUBLIC nombre, apellidos

PRIVATE sueldo

Operadores relacionales

Los operadores se utilizan en tiempo de ejecución para unir dos o más elementos del lenguaje.

Operadores aritméticos

() Agrupa subexpresiones

** , ^ Exponenciación

*, / Multiplicación y división

% Módulo (resto de la división)

+, - Suma y resta

Operadores relacionales

< Menor que

> Mayor que

<= Menor o igual que

>= Mayor o igual que

= igual a

<>, #, !=Distinto de

\$ Contenido en

Ahora te daremos unos ejemplos con funciones en Visual FoxPro.

* Declaración de variables

cadena1 = SPACE(15)

cadena2 = SPACE(15)

número = 0

cuenta = 0

* Asignación de datos

cadena1 = "abcd efgh ijkl"

* utilizando la función LEN para contar el número de la cadena

? LEN(Cadena1) && imprime en pantalla 14

? Cadena2= SUBSTR(cadena1,1,4) && imprime en pantalla abcd

? DTOC(DATE())

? SIN (0)

Índices

Cuando se insertan registros en una tabla estos no se encuentran ordenados, para ello es necesario utilizar comandos que nos permitan ordenar los registros de acuerdo a la información que necesitamos.

Comando SORT

Clasifica un archivo de base de datos

SINTAXIS

SORT TO <archivo> ON <campo> [/A] [/D] [/C]

[,<campo2> [/A] [/D] [/C]..]

[ASCENDING | DESCENDING]

[<alcances>]

[FOR <expresión lógica1>] [WHILE <expresión lógica2>]

[FIELDS <lista de campos>]

EJEMPLO:

USE persona

SORT TO persona_ordenada ON nombre

USE persona_ordenada

Persona	persona_ordenada
Juan Martínez	Eduardo Garay
Roxana Cuellar	Juan Martínez
Margarita Rosales	Margarita Rosales
Pedro López	Pedro López
Eduardo Garay	Roxana Cuellar

Comando INDEX ON

Crea un índice en un archivo de índices compuestos o bien un archivo de índice simple.

SINTAXIS

INDEX ON campo TO nombre de archivo IDX | TAG etiqueta

[OF archivo CDX][FOR expresión lógica] [COMPACT][ASCENDING | DESCENDING][CANDIDATE

[UNIQUE][ADDITIVE]

EJEMPLO:

USE persona

INDEX ON nombre TO persona.idx

El comando INDEX ON crea un archivo índice llamado persona.idx, el cual tendrá el índice de el archivo persona.

Una mejor forma de crear un índice y en el cual Fox trabaja a una mejor capacidad es creando un índice compuesto.

INDEX ON nombre TAG persona

De la forma anterior se crea un archivo índice persona.CDX.

También se pueden crear índices combinados con dos o más campos por ejemplo: INDEX ON UPPER(LEFT(apellido,10) + LEFT(nombre,10)) TO nombre.

Practica 2

1. Abre la tabla de inventario.

1. Escribe la siguiente instrucción SORT ON codigo TO inventarioordenado.
2. Aplica un BROWSE y abre la nueva tabla y BROWSE, compara ambas.
3. Cierra las tablas en uso escribiendo la siguiente sentencia CLOSE ALL.
4. Haremos una copia del archivo inventario.dbf COPY FILE inventario.dbf TO inventa.dbf.
5. Ahora ordenaremos la tabla por medio de un índice escribiendo la siguiente sentencia INDEX ON CODIGO TO CODIGO; le estas diciendo que el índice va ha ser por medio de código y que el archivo índice se llamará CODIGO solo que este con extensión IDX, visualiza el archivo escribiendo DIR *.IDX.
6. Ahora cierra la tablas. Y vuelve a abrirla escribiendo USE INVENTARIO INDEX CODIGO.
7. Busca un registro haciendo uso del comando SEEK, de la forma siguiente SEEK <expresión> por ejemplo SEEK "MO01"; recuerda si el campo es de tipo numérico no se le colocan comillas.
8. Ahora DISPLAY, y veras que te aparece el registro que buscaste por medio del índice.

Ejercicio:

Crema una tabla que guarde los datos de tus compañeros y pon un índice por el campo que creas conveniente y práctica lo anterior.

También puedes indexar una tabla, desde el generador de tablas de FoxPro marcando la casilla que se muestra en la figura:

Primero tienes que seleccionar el campo que servirá como índice, luego selecciona la ficha de modo que te aparezca la siguiente pantalla:

Tienes que elegir el tipo de índice el cual le indicara a la tabla que tipo de índice tiene. Los tipos de índices son los siguientes:

Principal: No admities claves duplicadas, actúa como referente a un registro de una tabla, de forma que no puede haber dos registros con la misma clave.

Candidato: Este índice tampoco admite claves duplicadas. Puede haber más de uno por tabla.

Único: En este índice sólo se guardan aquellos valores que sean distintos entre sí, pero sí admite registros con valores duplicados.

Regular: Es el índice típico. Admite claves duplicadas.

Relación de Tablas

Normalización

La única y más importante técnica en el desarrollo de base de datos es la capacidad de observar los datos y ver la base de la estructura de los mismos. Al tener varios datos desordenados y organizarlos en diferentes tablas relacionadas entre sí, a este proceso se le llama normalización.

Vincular tablas

SET RELATION TO

Crema una relación entre una base de datos padre, la que deberá estar abierta en el área de trabajo actualmente seleccionada, y una base de datos hija, ya abierta en otra área de trabajo.

SINTAXIS

SET RELATION TO [<expresión1> INTO <expresión numérica1>|<expresión carácter1>]

Ejemplo:

Para relacionar tablas tienes que tener do o más tablas abiertas en distintas áreas de trabajo y que por lo menos tengan un campo en común.

```
SELECT 1
```

```
USE empleado INDEX empleado.IDX
```

```
SELECT 2
```

```
USE departamento INDEX departamento.IDX
```

```
SELECT 1
```

```
SET RELATION TO cod_dpto INTO 2
```

En el ejemplo anterior empleado es la tabla padre y departamento la tabla hija, la relación de se da por medio del capo en común cod_dpto.

El generador de bases de dato

Almacena información sobre tablas, índices y otros temas relacionados. Un contenedor de base de datos en verdad es una tabla de FoxPro con la extensión .DBC. Sus campos memo relacionados están en una tabla del mismo nombre con la extensión .DCT.

Los DBC son esencialmente diccionarios de datos activos. Su representación visual de un modelo de datos de aplicación visual es precisa en la documentación y comunicación de las relaciones entre las tablas de sus aplicaciones.

Las relaciones más comunes entre tablas:

De uno a uno: Ejemplo de una llave primaria a otra llave primaria.

De uno a muchos: Ejemplo de una llave primaria a una regular o foránea.

Práctica 3.

Haciendo uso de los conocimientos adquiridos en Análisis de Sistemas, normaliza tu tabla de inventario y construye una base de datos del inventario que por lo menos tenga las tablas:

Productos, Proveedores y Marca. Construye el diagrama de Entidad - Relación con su cardinalidad y realiza la estructura de la base de datos en Visual FoxPro.

Después de que hayas relacionado la estructura de tu base de datos, has la siguiente prueba para verificar si la base de datos esta bien construida.

- a. Agrega a la tabla de Productos 20 registros, a la de proveedores 3 registros y a marcas 12 registros, tomando en cuenta que deben de tener datos en común como por ejemplo:

MO25 - Monitor de 14" - ACER - 1080.00 - Procom S.A.

- b. Para verificar la relación de los datos deberás realizar los siguientes pasos:

1. Elige el menú Ventana - Sesión de datos.
2. Te aparece la ventana de sesión de datos, luego elige el botón Abrir y selecciona una de las tablas.
3. Has lo mismo con las otras tablas.
4. Ya abiertas las tablas, selecciona la tabla de proveedores y has clic en el botón Relaciones, luego selecciona la tabla Marca y elige el campo con el cual armaras tu relación.
5. Si te aparece el cuadro expresiones, selecciona el campo de la tabla de la cual armaras la relación.
6. Luego has clic otra vez en la tabla proveedores y da clic en el botón Relaciones.
7. Repite los pasos anteriores solo que esta vez elige la tabla de producto, de modo que al finalizar te quede de la siguiente forma.
8. Ahora selecciona el botón Examinar para cada tabla y navega en la tabla proveedores, los datos de las otras tablas aparecerán relacionados con ella, de lo contrario revisa tu relación.
9. Prueba con la relación Producto - Marca - Proveedores, Marca - Productos - Proveedores.

3. Programación.

Conceptos.

Usualmente se agrupa el código fuente en archivos de programa. Un archivo de programa esta compuesto de varias sentencias, que incluyen comandos, funciones y estructuras de control de flujo.

El compilador traduce el contenido de estos archivos de texto ASCII a código interpretado. En cuanto el compilador ha traducido el código fuente crea un archivo con extensión .FXP que es el que va a utilizar a partir de ese momento. Cada vez que el programador introduce algún cambio en el archivo .PRG el compilador compara la fecha de ambos archivos y crea una nueva versión compilada del programa.

Haciendo el código legible:

- Espacios en blanco: Significan áreas vacías dentro del código de programa, también espacios, tabulaciones o líneas vacías.
- Sangrado: Sangrar es la mejor manera de optimizar el código. Es una práctica común sangrar dentro de comandos pares anidados, por ejemplo:

```
DO... ENDDO
IF... ENDIF
SCAN..ENDSCAN
DOCASE...ENDCASE
WITH...ENDWITH
```

- Mayúsculas y minúsculas: Muchos programadores utilizan mayúsculas y minúsculas para distinguir entre los elementos del programa; mayúsculas para los comandos y minúsculas para las variables.
- Comentarios: Los comentarios pueden existir en una línea precedidos por un asterisco y después de una línea de código, si están separados de && consecutivos.
- Continuar líneas largas de código: Puede continuar líneas largas de código en la siguiente línea insertando un ";" al final de la línea.

Una de las técnicas más útiles en la programación son los encabezados de programa, por ejemplo:

```
* Nombre del programa : nombre
* Autor : Héctor Hernández
* Propósito : Dibujar cuadros en la pantalla
* Fecha : 12/12/2001
```

Comandos de Asignación

Para crear variables de memoria y darles valores al mismo tiempo, escriba lo siguiente:

```
X=3
```

```
STORE "lunes" TO día
```

Para solamente crearlas:

```
PUBLIC a, b, c
```

```
PRIVATE x, y, z
```

```
LOCAL nombre, trabaja
```

```
DIMENSION nombres(3)
```

Declaración de Variables.

Las variables temporales se pueden declarar utilizando los comandos anteriores, únicamente, hay que tomar en cuenta la forma en que se van a declarar o escribir, como ejemplo: Nom_Empleado; NomEmpleado, si son variables de memoria se le puede anteponer una 'm ó x' de la forma siguiente, mNom_Empleado; xNom_Empleado, también se pueden declarar variables haciendo saber el tipo de datos que maneja por ejemplo: cNom_Empleado, nSueldo_Empleado; dFecha_Ingreso, etc.

Operadores.

Los operadores + y – son, por supuesto, usados para añadir y substraer números, pero también se pueden usar en campos de caracteres. El operador más eslabona dos expresiones de caracteres, campos o variables de memoria, de forma que Fred + Smith; producirá: FredSmith. El operador menos quita espacio rezagado.

Operadores cíclicos y ramificaciones.

Las operaciones cíclicas y de ramificación son los mecanismos para crear la lógica de un programa, ejemplo:

```
FOR I=1 TO 10
```

```
? I && Imprime los números del 1 al 10
```

```
ENDFOR
```

```
SELECT clientes
```

```
SCAN
```

```
? Clientes.nombre&&Lee un archivo desde el inicio hasta el final
```

```
ENDSCAN
```

```
SELECT clientes
```

```
GO TOP
```

```
DO WHILE NOT EOF()
```

```
? Clientes.nombre&& Lee un archivo desde el inicio hasta el final
```

```
SKIP
```

```
ENDDO
```

Puede salir de estas construcciones cíclicas en cualquier momento mediante el comando: EXIT.

La ramificación en VF es controlada con la construcción IF.. ENDIF, por ejemplo:

```
IF I >10
```

```
? [El contador excedió 10]
```

```
ENDIF
```

Los contadores se utilizan a menudo para salir de operaciones cíclicas:

```
L=1
```

```
DO WHILE L=LEN(nombrelista)
```

```
IF [CRUZ]$ nombrelista(L)
```

```
SEEK nombrelista(L)
```

```
IF nombre= [John]
```

```
DO prgnombre
```

```
LOOP
```

```
ENDIF
```

```
ENDIF
```

```
ENDDO
```

Se recomienda que cuando se utiliza un operador cíclico, un operador condicional, o cualquier otro operador que abra y cierre una estructura se escriba el comando primero y luego se realice el proceso, esto para no perderse en la programación.

Tipos de comandos

Vamos a dividir los comandos según su funcionalidad en los siguientes tipos:

1. Base de datos: Todos los que sirven para manipular bases de datos, tablas y campos.
2. Entorno: Entorno del sistema operativo y de Visual Fox Pro.
3. Acceso compartido a datos: Acceso multiusuario a tablas y base de datos.
4. Entrada y salida: Los que permiten la interfaz entre la computadora y el usuario.
5. Programación: Los que tienen relación con el entorno de desarrollo de aplicaciones.

Funciones

Son procesos internos programadores que pueden ser llamados desde cualquier punta de Visual Fox lo que diferencia de los mandatos de las funciones es que estas devuelven un valor. Aceptan

una serie de parámetros, y se reconocen por el uso de paréntesis para encerrar los argumentos, por ejemplo DATE() y TIME ().

Las funciones se dividen de la siguiente forma:

- Tipos de datos
- Funciones de manejo de cadenas de caracteres.
- Funciones de conversión de tipos
- Funciones de fecha y hora
- Funciones de manejo de valores numéricos
- Funciones definidas por los usuarios (FDU)

Practica 4

Realiza los siguientes programas utilizando programación simple (programas **.PRG**) y da solución a las siguientes situaciones.

1. Desplegar el valor a cobrar por llamadas telefónicas locales, si se cobra \$0.09 cada llamada, haciendo hasta 100 llamadas, \$0.08 por 200, \$0.07 por 300 y \$0.06 por más de trescientos, teniendo en cuenta también que por llamadas internacionales la tarifa es fija de \$1.93 si son diurnas y \$1.75 si son nocturnas, cada una de estas tarifas esta medida en minutos.
2. Calcular la utilidad que un trabajador recibe en el reparto de utilidades si éste se le asigna como porcentaje de un salario mensual que depende de su antigüedad en la empresa de acuerdo con esta tabla:

Tiempo Utilidad

Menos de un año 5%

de 1 hasta menos 2 años 7%

2 años a menos de 5 años 10%

de 5 años a más 15%

3. Determinar la cantidad de dinero que recibirá un trabajador por concepto de las horas trabajadas en una empresa, sabiendo que cuando las horas extras de trabajo exceden de 40, el resto se considera horas extras y que éstas se pagan al doble de una hora normal cuando no exceden de 8; si las horas extras exceden de 8 se pagan al doble más el 70% de la hora normal, las horas extras no exceden de 55.
4. Realice un programa en el cual pida el nombre completo de una persona y este indique cuantas vocales y cuantas consonantes tiene.
5. Realice un programa o diferentes programas que permita un mantenimiento de una tabla de direcciones.

Programación orientada a objetos

La POO intenta ser un mejor sistema para el desarrollo de aplicaciones. Como toda técnica de programación, si se hace mal puede ser desastrosa.

Una de las mejoras que se tiene con la POO es el permitir afrontar programas más complejos y de mayor tamaño con menor esfuerzo. El hecho de trabajar con pequeños elementos bien definidos, como son los objetos, nos permite aislar cada componente de la aplicación, del resto y de esa forma aprovechar en mayor medida nuestro esfuerzo.

Elementos de la POO

Clase y objeto: la clase es la generalización de los objetos y los objetos son la concreción de la clase. Como ejemplo podemos observar los botones de un entorno gráfico, donde definimos que la

clase botón es la generalización de las propiedades y comportamientos de todos los botones de los entorno gráficos.

Definir una clase: las clases son la descripción de los elementos comunes de los objetos que generalizan. Así las clases se definen y pueden ser usadas para crear innumerables objetos de este tipo. Para definir una clase utilizaremos una sencilla sintaxis de VFP.

DEFINE CLASS NombreClase AS Clasebase

Para poder utilizar esta definición debemos incluirla en un fichero .PRG y cargado con SET PROCEDURE TO o bien incluirla al final de nuestro fichero .PRG.

Crear un objeto: ya podemos crear objetos basado en esta clase, para ello utilizaremos la siguiente expresión:

Variable1 = CREATEOBJECT("NombreClase")

Propiedades: Como hemos dicho, las propiedades son los datos que manejan las clases. Estas propiedades se declaran en la definición de la clase y permanecen en todo momento asociados a los objetos CREADOS bajo esa clase.

Métodos: el otro elemento característico de una clase son los métodos. Los métodos son acciones que pueden realizar los objetos, es decir, son funciones o procedimientos asociados a este tipo objeto.

Mensajes: Cuando llamamos a un método de un objeto se dice que estamos enviando un mensaje al objeto para que realice una determinada acción.

Operador this: Cuado vamos a utilizar una propiedad o un método de la clase, debemos anteponer al operador punto el operador THIS, para indicar que se trataran las propiedades del objeto que recibe el mensaje, es decir, que ha sido invocado, y no para otro.

Las propiedades mantienen valores diferentes para cada uno de los objetos, pero los métodos comparten su código entre todos los objetos de una clase. Un método varía en la medida que las propiedades del objeto que lo llama son diferentes, por ello es tan importante el operador THIS.

Ocultación: Una de las mejoras que implementa la POO, es la posibilidad de limitar el acceso a determinadas propiedades o métodos. Con ello conseguimos que la utilización de la clase se haga de forma ordenada.

Las propiedades o métodos protegidos sólo son utilizables desde los métodos pertenecientes a esta clase y no pueden usarse directamente por otros programas.

Al igual que podemos proteger propiedades, podemos proteger métodos. De esta forma podemos definir métodos que sólo sean usados por otros métodos de la clase y no puedan ser invocados a partir de los objetos de esta clase. Para ello basta colocar la cláusula PROTECTED antes de PROCEDURE.

Polimorfismo: Cuando realizamos programación estructurada debemos tener cuidado de no llamar con el mismo nombre a dos variables o a dos procedimientos, sin embargo en la POO podemos llamar a un método o a una propiedad de una clase de igual forma que un método o propiedad de otra. Esta característica es lo que se denomina polimorfismo.

Eventos: Existe una serie de métodos especiales, que normalmente no se ejecutan por ser invocados de forma explícita, como los que hemos definido hasta ahora, sino que por denominarse de una forma determinada son lanzados cuando "pasa algo", es decir, cuando se produce un evento. Estos eventos pueden ser un clic, el movimiento del ratón, una pulsación de tecla, etc.

Conclusiones

La POO es un conjunto de conceptos interrelacionados que difícilmente se entiende unos sin los otros. La POO está aquí y no deberíamos ignorarla por más tiempo. Posiblemente no es necesario este tipo de programación, pero es realmente muy recomendable, es seguro que no soluciona todos los problemas, pero es mucho más sencillo el desarrollo, tendremos que esforzarnos un poco al principio, pero nuestro esfuerzo se verá sobradamente recompensado. En definitiva la PROGRAMACIÓN ORIENTADA A OBJETO es una mejor forma de programar.

Ejemplo:

■ DEFINE EL OBJETO formu

formu = CREATEOBJECT("Ejemplo")

- DEFINE EL OBJETO boton

```
formu.ADDOBJECT("boton","boton")
formu.SHOW
READ EVENT
```

- CLASE DEL OBJETO formu

```
DEFINE CLASS ejemplo AS FORM
MOVABLE=.F.
CLOSABLE=.F.
CAPTION="EJEMPLO DE POO"
AUTOCENTER=.T.
ENDDFINE
```

- CLASE DEL OBJETO boton

```
DEFINE CLASS boton AS COMMANDBUTTON
VISIBLE=.T.
CAPTION="PRESIONE AQUI"
LEFT=125
TOP=65
HEIGHT=100
PROCEDURE CLICK
IF MESSAGEBOX("ACABAS DE UTILIZAR POO",0+48)=1
CLEAR EVENT
ENDIF
ENDDFINE
```

- Fin del programa

4. Formularios

Conceptos

Objeto: Una instancia de una clase que combina datos y procedimientos. Por ejemplo, un control de un formulario en ejecución es un objeto.

Formulario: Se utilizan para visualizar e introducir cualquier tipo de información y son el medio de comunicación entre el usuario y los datos.

Propiedades: Un atributo de un control, campo u objeto de base de datos que se establece para definir una de las características del objeto o un aspecto de su comportamiento. Por ejemplo, la propiedad Visible afecta a la visibilidad en tiempo de ejecución de un control. Puede cambiar los valores de las propiedades de objeto mediante la ventana Propiedades.

Evento: Una acción, reconocida por un objeto, para la cual puede escribir código de respuesta. Los eventos pueden estar generados por una acción del usuario, como hacer clic con el mouse o presionar una tecla, por código de programa o por el sistema, como ocurre con los cronómetros.

Método: Una acción que un objeto es capaz de realizar. Por ejemplo, los cuadros de lista tienen métodos llamados AddItem, RemoveItem y Clear para mantener el contenido de las listas.

Barra de herramientas: Una serie de botones en los que se hace clic para realizar tareas frecuentes. Las barras de herramientas pueden flotar en su propia ventana o bien puede acopladas en los extremos superior, inferior o laterales de la ventana principal de Visual FoxPro. Puede personalizar las barras de herramientas suministradas con Visual FoxPro, así como crear barras de herramientas propias mediante la clase de base ToolBar incluida con Visual FoxPro.

Objetos y propiedades básicas

Form

El formulario es un objeto de tipo contenedor.

Los conjuntos de formularios acogen a uno o más formularios, o incluso barra de herramientas. Ala hora de ejecutar el conjunto de formularios, se activan todos los formularios definidos en el mismo.

Propiedad	Uso
Caption	Título de la ventana o forma
Autocenter	Centra el formulario
ShowTips	Para que se muestren los Tooltips
Closable	Visualiza el botón cerrar de la ventana
MaxButton	Visualiza el botón maximizar
MinButton	Visualiza el botón minimizar
ControlBox	Visualiza el menú de control de la ventana
Eventos o Métodos	Uso
Init	Se dispara a la hora que se inicializan todos los objetos del formulario
Load	Se dispara antes que se inicialicen los objetos del formularios (antes que el init)
Destroy	Se dispara al cerrar el formulario
Unload	Se dispara después de Destroy

Viñetas y cuadros de texto

Las viñetas o Labels se utilizan para escribir etiquetas a los objetos.

Propiedad	Uso
Caption	Título de la ventana o forma
Alignment	Especifica la alinación del texto
Name	Nombre del objeto
BackColor	Color de fondo
ForeColor	Color de la letra
Evento o Método	Uso
Click	Se dispara al dar clic en el objeto
MouseDown	El evento MouseDown se desencadena al presionar un botón del mouse
MouseUp	Ocurren cuando el usuario presiona (MouseDown) o suelta (MouseUp) un botón del mouse

Los cuadros de texto sirven para capturar los datos de los campos.

Propiedades	Uso
ControlSource	Es la variable o nombre del campo al cual se hace referencia
Value	Contenido de la variable o campo referenciado al ControlSource
Visible	Si el control es visible al usuario

Enabled	Si el control esta habilitado o no
Evento o Método	Uso
Click	Se dispara al dar clic en el objeto
GotFocus	Cuando el control toma el enfoque
LostFocus	Cuando el control pierde el enfoque

Botones de comando

Un control que está asociado a un comando. Cuando hace clic en el botón de comando en tiempo de ejecución, el comando asociado al botón se ejecuta.

Propiedades	Uso
Caption	Etiqueta del comando
Visible	Si el control es visible al usuario
Enabled	Si el control esta habilitado
Evento o Método	Uso
Click	Se dispara al dar clic en el objeto
SetFocus	El cursor se va al objeto
LostFocus	Cuando el control pierde el enfoque
Refresh	Refresca el objeto
Init	Se dispara a la hora que se se inicializa el objeto

Botones de opción y casillas de verificación

Un control `OptionButton` muestra una opción que se puede activar o desactivar.

Propiedades	Uso
ControlSource	Es la variable o nombre del campo al cual hace referencia
Value	Contenido de la variable o campo referenciado en <code>ControlSource</code>
Visible	Si el control es visible al usuario
Enabled	Si el control esta habilitado o no
Evento o Método	Uso
InteractiveChange	Se dispara cuando modificamos el campo
Init	Se dispara a la hora en que se inicializa el objeto
Click	Se dispara al dar click en el objeto
Valid	Se dispara después de presionar <code>Enter</code> o <code>Tab</code> , se utiliza para validar
Refresh	Refresca el objeto

Un control `CheckBox` muestra una X cuando está activado; la X desaparece cuando el control `CheckBox` se desactiva. Utilice este control para ofrecer al usuario una opción de tipo Verdadero o Falso o Sí o No. Puede usar controles `CheckBox` en grupos para mostrar múltiples opciones entre las cuales el usuario puede seleccionar una o más. También puede establecer el valor de `CheckBox` mediante programación con la propiedad `Value`.

Listas y cuadros combinados

Un control ListBox muestra una lista de elementos entre los cuales el usuario puede seleccionar uno o más. Si el número de elementos supera el número que puede mostrarse, se agregará automáticamente una barra de desplazamiento al control ListBox.

Si no se selecciona ningún elemento, el valor de la propiedad ListIndex será -1. El primer elemento de la lista es ListIndex 0 y el valor de la propiedad ListCount siempre es uno más que el mayor valor de ListIndex.

Los cuadros de lista y los cuadros combinados presentan al usuario una lista de opciones. De forma predeterminada, las opciones se muestran verticalmente en una única columna, aunque también puede establecer múltiples columnas. Si el número de elementos supera a los que se pueden mostrar en el cuadro combinado o el cuadro de lista, aparecen automáticamente barras de desplazamiento en el control. El usuario puede entonces desplazarse por la lista hacia arriba o hacia abajo o de izquierda a derecha.

Un cuadro combinado aúna las características de un cuadro de texto y un cuadro de lista. Este control permite al usuario seleccionar opciones si escribe texto en el cuadro combinado o selecciona un elemento de la lista.

A diferencia de otros controles que contienen un único valor, por ejemplo la propiedad Caption de una etiqueta o la propiedad Text de un cuadro de texto, los cuadros de lista y los cuadros combinados contienen múltiples valores o una colección de valores. Tienen métodos integrados para agregar, quitar y recuperar valores de sus colecciones en tiempo de ejecución. Para agregar varios elementos a un cuadro de lista llamado Lista1, el código sería como el siguiente:

```
Lista1.AddItem "París"
```

```
Lista1.AddItem "Nueva York"
```

```
Lista1.AddItem "San Francisco"
```

Los cuadros de lista y los cuadros combinados son una manera efectiva de presentar al usuario gran cantidad de opciones en un espacio limitado.

Propiedades	Uso
ControlSource	Variable a que hace referencia (campo donde se almacena el valor de la lista)
RowSource	Tabla o query de los elementos de la lista
rowSourceType	Tipo de la fuente (si es tabla o query)
Visible	Si el control es visible al usuario
Enabled	Si el control esta habilitado o no
BoundColumn	Valor que retorna la lista (número de la columna de RowSource)
BoundTo	Especifica si el valor de la lista esta determinado por los valores o por la propiedad ListIndex
Evento o Método	Uso
Click	Se dispara al dar clic en el objeto
SetIFocus	El cursor se va al objeto
LostFocus	Cuando el control pierde el enfoque
Refresh	Refresca el objeto
Init	Se dispara a la hora que se se inicializa el objeto

Cuadrícula

Un control que crea una cuadrícula y que le permite presentar los datos en un formato tabular. Los controles de tipo cuadrícula contienen encabezados de columna, columnas y controles de columna.

Propiedades	Uso
RecordSource	Nombre de la tabla
Visible	Si el control es visible al usuario
Enabled	Si el control esta habilitado o no
ColumnCount	Número de columnas del Grid
DeleteMark	Permite registro para borrar (Aparece en la parte izquierda del Grid)
Evento o Método	Uso
Click	Se dispara al dar clic en el objeto
SetIFocus	El cursor se va al objeto
LostFocus	Cuando el control pierde el enfoque
Refresh	Refresca el objeto
Init	Se dispara a la hora que se se inicializa el objeto

Imágenes, controles OLE, timer, Marcos de página

Un control que le permite incluir imágenes en un formulario.

Hipervínculo con cualquier tipo de aplicación.

Un control Timer puede ejecutar código a intervalos periódicos produciendo un evento Timer.

Un marco de página es un objeto contenedor que contiene páginas. A su vez, las páginas contienen controles. Las propiedades pueden establecerse a nivel de marco de página, de página o de control.

Uso de Clases

Los diseñadores ActiveX pueden proporcionar interfaces visuales para tareas que, de otro modo, requerirían una gran cantidad de código. Por ejemplo, el diseñador UserConnection incluido en la Edición Empresarial de Visual Basic proporciona herramientas visuales para definir consultas de bases de datos complejas. En tiempo de ejecución, dichas consultas se pueden invocar con muy poco código.

Similitudes entre los diseñadores ActiveX y los diseñadores integrados

Los diseñadores ActiveX son como los diseñadores de formularios en los siguientes aspectos:

- Los diseñadores ActiveX producen clases a partir de las cuales puede crear objetos. Estas clases aparecen en la ventana Proyecto, igual que las clases de formulario.
- Las clases creadas con un diseñador ActiveX tienen sus propios módulos de código, en los que puede escribir código para los procedimientos de evento proporcionados por el diseñador.
- Puede personalizar una clase si agrega propiedades, métodos y eventos a los proporcionados por el diseñador ActiveX.
- Los objetos creados a partir de las clases diseñadas pueden tener características diferentes en tiempo de diseño y en tiempo de ejecución.
- La ventana de diseño del diseñador ActiveX está totalmente integrada en el entorno de desarrollo. Puede modificar su tamaño y configurarla de la misma forma que las ventanas de diseño incorporadas.
- Puede agregar al proyecto tantas instancias de un diseñador ActiveX como considere necesario, de la misma forma que puede agregar tantos diseñadores de formularios como quiera.

Práctica 5

1. Diseña un formulario que represente y baucher de notas y que al escribir las notas calcule los promedios finales. (Puedes mejorarlo si los datos los vas almacenando en una tabla).
2. Realiza un programa que calcule el pago de una planilla donde el pago mínimo es de \$240.00, construye la tabla y has que los datos se almacenen en ella, toma en consideración, el pago por horas, horas extras, los descuentos necesarios y las comisiones tomando en consideración la siguiente tabla:

Si las ventas < \$1700.00 no hay comisión

Si las ventas >=\$1700.00 y <\$2500.00 la comisión es del 3%

Si las ventas >=\$2500.00 y <\$3200.00 la comisión es del 5%

Si las ventas >=\$3200.00 y <\$3700.00 la comisión es del 7%

Si las ventas >=\$3700.00 la comisión es del 10%

3. Utilizando tu base de datos de Inventario construye el mantenimiento respectivo sin utilizar el WIZARD.

Combinación de formularios y datos

La idea fundamental del trabajo con formularios es la combinación que estos hacen con las bases de datos, para ello, un formulario se puede combinar desde una tabla, hasta, varias de ellas. Para lograr esta combinación los formularios hacen uso de los objetos, los cuales se asocian a los datos, el objeto guarda la referencia a las tablas, y cuando el formulario se ejecuta, se abren específicamente todas las tablas asociadas al entorno de datos.

El entorno de datos es donde se definen las tablas relacionadas y sus ordenes establecidos.

Para lograr un formulario final se hace necesario de conocer los siguientes elementos:

Práctica 6

Crema una carpeta llamada Sistema de clientes, dentro de ella crea otras subcarpetas y nómbralas así: datos, formularios, gráficos e informes.

Utilizando el método de carpetas, te aseguras que tu proyecto se vea ordenado y clasificado.

Ahora cada objeto que crees de este ejemplo asegurate que lo guardas en el lugar indicado de lo contrario no funcionará debidamente.

Crema el proyecto

Selecciona Archivo / Nuevo / Proyecto

Guárdalo en la carpeta Sistema de clientes con el nombre de Clientes.

Crema la base de datos

De tu proyecto elige la ficha datos / Nuevo

Guárdalo en la carpeta Sistema de clientes / datos con el nombre de Clientes

Agrega tablas

Ahora agregaras las tablas correspondientes a la base de datos, selecciona Agregar nueva tabla y adiciona cada tabla que a continuación te damos, ten cuidado de grabarlas con el mismo nombre y con los nombres de los campos iguales.

Tabla: Clientes Índice primario: cod_cliente

Tabla: Distribuidor Índice primario: cod_distri

Tabla: Departamento Índice pprimario: cod_dpto

Copia los gráficos necesarios en la carpeta Gráficos.

Relaciona las tablas.

Cuando a cabes de crear las tablas relaciona las tablas de modo que te queden de la siguiente forma.

Diseño de formulario.

Ahora, haremos el diseño del formulario, puedes hacerlo de dos formas, la primera es abriendo el entorno de datos y arrastrando los campos hasta el formulario, y la segunda colocando cada uno de los objetos, luego tu cambias la propiedad ControlSource colocando el nombre del campo al que hace referencia, el diseño del formulario puede ser el siguiente, pero tu puedes hacer el tuyo. Aseguraté que las tablas en el entorno de datos se encuentren bien relacionadas, es decir: entre la relación Clientes – Departamento la propiedad del entorno de datos ChildAlias = Departamento y ParentAlias = Clientes y entre la relación Departamento – Distribuidor las propiedades serian ChildAlias = Distribuidor y ParentAlias = Departamento. Si tienes alguna duda pregunta a tu instructor.

Cambia propiedades necesarias.

Cambia las propiedades que te son útiles, por ejemplo FontName, FontSize, BackColor y para los cuadros de texto y cuadros de edición cambia la propiedad Enabled, esta servirá para no poder escribir mientras no se le de la orden de agregar o modificar registros.

Crearemos una barra de herramientas.

Haremos la barra de cursuras, para eso tu instructor te copiará los gráficos necesarios en tu disco.

1. Primero en la parte de abajo del formulario, coloca un commandgroup y cambia la propiedad ButtonCount con el valor de 4, para que aparezcan cuatro botones de comando, has clic derecho sobre ella y selecciona del menú contextual que aparece la opción modificar, el cuadro se sobraré de color verde, esto te indica que puedes modificar los botones que están en u interior.
2. Cambia las propiedad name de cada uno de ellos por: primero, anterior, siguiente y último respectivamente y la propiedad Caption déjala vacía (o sea que borra las leyendas).
3. Ahora cambia la propiedad Picture y busca el gráfico que pertenece a cada botón, de modo que te quede de la siguiente forma:

Programa los botones de la barra.

Botón primero:

GO TOP

THISFORM.REFRESH

Botón anterior:

IF .NOT. BOF()

SKIP -1

ELSE

GO BOTTOM

ENDIF

THISFORM.REFRESH

Botón Siguiente:

IF .NOT. EOF()

SKIP 1

ELSE

GO TOP

ENDIF

THISFORM.REFRESH

Botón último:

GO BOTTOM

THISFORM.REFRESH

Agrega un botón cerrar

Ahora inserta un botón para cerrar la ventana que estas programando, puedes hacerlo con texto o con gráficos, programa lo siguiente:

THISFORM.RELEASE

Agrega un botón para buscar registros

Vamos a agregar un botón para que busque registros por medio del campo índice código del cliente, para ello inserta un contenedor y dentro del un botón de comandos y un cuadro de texto, modifica las propiedades necesarias de modo que te quede como el siguiente ejemplo:

```
buscar= 0
buscar=VAL(THISFORM.CONTAINER1.txtbuscar.VALUE)
SEEK buscar ORDER TAG cod_cliente;
OF 'C:\Hector Hernandez\Sistema de Clientes\Datos\clientes.cdx'
IF !FOUND()
ctexto = 'Registro no encontrado. ¿Desea buscar de nuevo?'
ntipodedialogo = 4 + 32 + 256
ctitulo = 'Mensaje'
* 4 = Botones Sí y No.
* 32 = Icono del signo de interrogación.
* 256 = El segundo botón es el predeterminado.
nrespuesta = MESSAGEBOX(ctexto, ntipodedialogo, ctitulo)
DO CASE
CASE nrespuesta = 6
THISFORM.CONTAINER1.txtbuscar.SETFOCUS()
CASE nrespuesta = 7
THISFORM.cmdsalir.SETFOCUS()
ENDCASE
ELSE
THISFORM.REFRESH
ENDIF
```

Agrega la barra de mantenimiento.

Has un diseño similar al siguiente dentro de un grupo de botones:

Luego, inserta dos botones más arriba de cmdagregar y cmdeditar, de modo que te quede de la siguiente forma:

Ha estos botones programa lo siguiente:

```
Cmdagregar
THISFORM.txtcod_cliente.ENABLED=.t.
THISFORM.txtnom_cliente.ENABLED=.t.
THISFORM.txtape_cliente.ENABLED=.t.
THISFORM.txtnom_negocio.ENABLED=.t.
THISFORM.edtdir_cliente.ENABLED=.t.
THISFORM.txtcod_dpto.ENABLED=.t.
THISFORM.txttele_cliente.ENABLED=.t.
THISFORM.txtfax_cliente.ENABLED=.t.
THISFORM.txtno_cuenta.ENABLED=.t.
THISFORM.commandgroup1.ENABLED=.f.
THISFORM.container1.ENABLED=.F.
THISFORM.commandgroup2.ENABLED=.F.
THISFORM.cmdsalir.ENABLED=.F.
THISFORM.cmdguardar.VISIBLE=.T.
GO BOTTOM
THISFORM.REFRESH
contador=THISFORM.txtcod_cliente.VALUE
APPEND BLANK
THISFORM.txtcod_cliente.value=contador+1
THISFORM.txtnom_cliente.SETFOCUS
THISFORM.REFRESH
Cmdguardar
ctexto = '¿Esta seguro de grabar el registro?'
ntipodedialogo = 4 + 32 + 256
ctitulo = 'Guardar'
* 4 = Botones Sí y No.
```

```

* 32 = Icono del signo de interrogación.
* 256 = El segundo botón es el predeterminado.
nrespuesta = MESSAGEBOX(ctexto, ntipodedialogo, ctitulo)
DO CASE
CASE nrespuesta = 6
THISFORM.REFRESH
CASE nrespuesta = 7
DELETE
PACK
ENDCASE
THISFORM.txtcod_cliente.ENABLED=.f.
THISFORM.txtnom_cliente.ENABLED=.f.
THISFORM.txtape_cliente.ENABLED=.f.
THISFORM.txtnom_negocio.ENABLED=.f.
THISFORM.edtdir_cliente.ENABLED=.f.
THISFORM.txtcod_dpto.ENABLED=.f.
THISFORM.txttele_cliente.ENABLED=.f.
THISFORM.txtfax_cliente.ENABLED=.f.
THISFORM.txtno_cuenta.ENABLED=.f.
THISFORM.commandgroup1.ENABLED=.t.
THISFORM.container1.ENABLED=.t.
THISFORM.commandgroup2.ENABLED=.t.
THISFORM.cmdsalir.ENABLED=.t.
THISFORM.cmdguardar.VISIBLE=.f.
THISFORM.REFRESH
Cmdborrar
mensaje=MESSAGEBOX("¿Seguro de eliminar el registro?"; 4+32+256,"Eliminar registros")
DO CASE
CASE mensaje = 6
DELETE
PACK
GO TOP
THISFORM.REFRESH
CASE mensaje = 7
THISFORM.REFRESH
ENDCASE
Cmdeditar
THISFORM.txtnom_cliente.ENABLED=.t.
THISFORM.txtape_cliente.ENABLED=.t.
THISFORM.txtnom_negocio.ENABLED=.t.
THISFORM.edtdir_cliente.ENABLED=.t.
THISFORM.txtcod_dpto.ENABLED=.t.
THISFORM.txttele_cliente.ENABLED=.t.
THISFORM.txtfax_cliente.ENABLED=.t.
THISFORM.txtno_cuenta.ENABLED=.t.
THISFORM.commandgroup1.ENABLED=.f.
THISFORM.container1.ENABLED=.F.
THISFORM.commandgroup2.ENABLED=.F.
THISFORM.cmdsalir.ENABLED=.F.
THISFORM.cmdguardare.VISIBLE=.T.
THISFORM.txtnom_cliente.SETFOCUS
THISFORM.REFRESH
Cmdgrabare
THISFORM.txtcod_cliente.ENABLED=.f.
THISFORM.txtnom_cliente.ENABLED=.f.
THISFORM.txtape_cliente.ENABLED=.f.

```

THISFORM.txtnom_negocio.ENABLED=.f.
 THISFORM.edtdir_cliente.ENABLED=.f.
 THISFORM.txtcod_dpto.ENABLED=.f.
 THISFORM.txttele_cliente.ENABLED=.f.
 THISFORM.txtfax_cliente.ENABLED=.f.
 THISFORM.txtno_cuenta.ENABLED=.f.
 THISFORM.commandgroup1.ENABLED=.t.
 THISFORM.container1.ENABLED=.t.
 THISFORM.commandgroup2.ENABLED=.t.
 THISFORM.cmdsalir.ENABLED=.t.
 THISFORM.cmdguardare.VISIBLE=.f.
 THISFORM.REFRESH

Ahora cambiaremos algunas propiedades indispensables, al formulario cambia ShowWIndor=2- como formulario de nivel superior, ShowTips=.t., y ahora cambia algunas propiedades ToolTipsText para que aparezcan algunas indicaciones al colocar el puntero del ratón arriba del control.

Agregaremos otro formulario.

En el generador de proyectos diseñaremos otro formulario llamado departamento:

Este formulario será pequeño, trata que en la posición que lo diseñes sea donde en el formulario clientes vaya el departamento del cliente, agrega un control Grid (Cuadrícula), has clic derecho sobre ella y selecciona la opción Generador.

Te aparecerá la siguiente pantalla:

Selecciona la tabla Departamento, selecciona los campos, luego pasa a la ficha estilo y escoge el estilo que quieras, selecciona la ficha distribución, escribe los títulos que allí te piden y por último la ficha relación, donde escribirás ninguna, por último da clic ha aceptar y automáticamente se te generará un grid con sus especificaciones.

Luego a ese formulario cambia la propiedad ShowWIndor=1- en formulario de nivel superior y grábalo.

Modifica el cuadro de edición.

Entra al formulario clientes, selecciona el formulario la propiedad INIT y copia:

PUBLIC CODEPAR

CODEPAR=0

contador=0

Luego al cuadro de edición en la propiedad LOSTFOCUS escribe:

DO FORM 'C:\Hector Hernandez\Sistema de Clientes\formularios\departamento'

Luego en la propiedad GOTFOCUS del objeto txtcod_dpto escribe lo siguiente:

THIS.VALUE=CODEPAR

THISFORM.REFRESH

Ahora puedes probar tu programa.

Usando el asistente de formularios.

Con el asistente, crea el mantenimiento de distribuidores.

5. Informes

Utilice el Diseñador de informes para crear y modificar informes.

Para crear rápidamente un diseño sencillo de informe, elija Informe rápido del menú [Informe](#).

Informe rápido le solicita la entrada de los campos y el diseño que desea para su informe.

Opciones de la ventana

Bandas

De forma predeterminada, el Diseñador de informes muestra tres [bandas](#): [Encabezado de página](#), [Detalle](#) y [Pie de página](#). En la parte inferior de cada banda hay una barra separadora. El nombre de la banda aparece en la barra gris junto a una flecha azul, que indica que la banda está encima, no debajo, de la barra gris.

Puede agregar las siguientes bandas a su informe.

Banda	Se imprime	Contenido típico
Encabezado de columna	Una vez por columna	Título de columna

Pie de columna	Una vez por columna	Resumen, totales
Encabezado de grupo	Una vez por grupo	Precede los datos siguientes
Pie de grupo	Una vez por grupo	Valores calculados para grupos de datos
Título	Una vez por informe	Título Fecha o número de página Logotipo de la compañía Cuadro alrededor del título
Resumen	Una vez por informe	Totales Texto como Suma total

Regla

El Diseñador de informes tiene una regla vertical y otra horizontal que puede utilizar para colocar de forma más precisa los objetos en las bandas. Utilice las reglas junto con el comando [Ver](#) del menú Mostrar posición para ayudarle en la colocación de los objetos.

La escala de la regla viene determinada por las configuraciones de medidas de su sistema. Puede cambiar de la escala predeterminada del sistema (pulgadas o centímetros) a píxeles desde dentro de Visual FoxPro. Si desea cambiar al valor predeterminado del sistema, cambie la configuración de medidas de su sistema operativo.

Para cambiar la escala de la regla

Cambie esta unidad a píxeles de la forma siguiente:

1. En el menú Formato, elija Configurar cuadrícula. Se muestra el cuadro de diálogo [Configurar cuadrícula](#).
2. En el cuadro Escala de la regla, seleccione Píxeles y elija Aceptar.

La escala de la regla se configura a píxeles y el indicador de posición de la barra de estado (si está activado Mostrar posición en el menú Ver) también muestra las posiciones en píxeles.

Trabajo enviado por:

Rotce hHernandez

rotcehHernandez@terra.com