FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA

A LOS ASISTENTES A LOS CURSOS

Las autoridades de la Facultad de Ingeniería, por conducto del jefe de la División de Educación Continua, otorgan una constancia de asistencia a quienes cumplan con los requisitos establecidos para cada curso.

El control de asistencia se llevará a cabo a través de la persona que le entregó las notas. Las inasistencias serán computadas por las autoridades de la División, con el fin de entregarle constancia solamente a los alumnos que tengan un mínimo de 80% de asistencias.

Pedimos a los asistentes recoger su constancia el día de la clausura. Estas se retendrán por el periodo de un año, pasado este tiempo la DECFI no se hará responsable de este documento.

Se recomienda a los asistentes participar activamente con sus ideas y experiencias, pues los cursos que ofrece la División están planeados para que los profesores expongan una tesis, pero sobre todo, para que coordinen las opiniones de todos los interesados, constituyendo verdaderos seminarios.

Es muy importante que todos los asistentes llenen y entreguen su hoja de inscripción al inicio del curso, información que servirá para integrar un directorio de asistentes, que se entregará oportunamente.

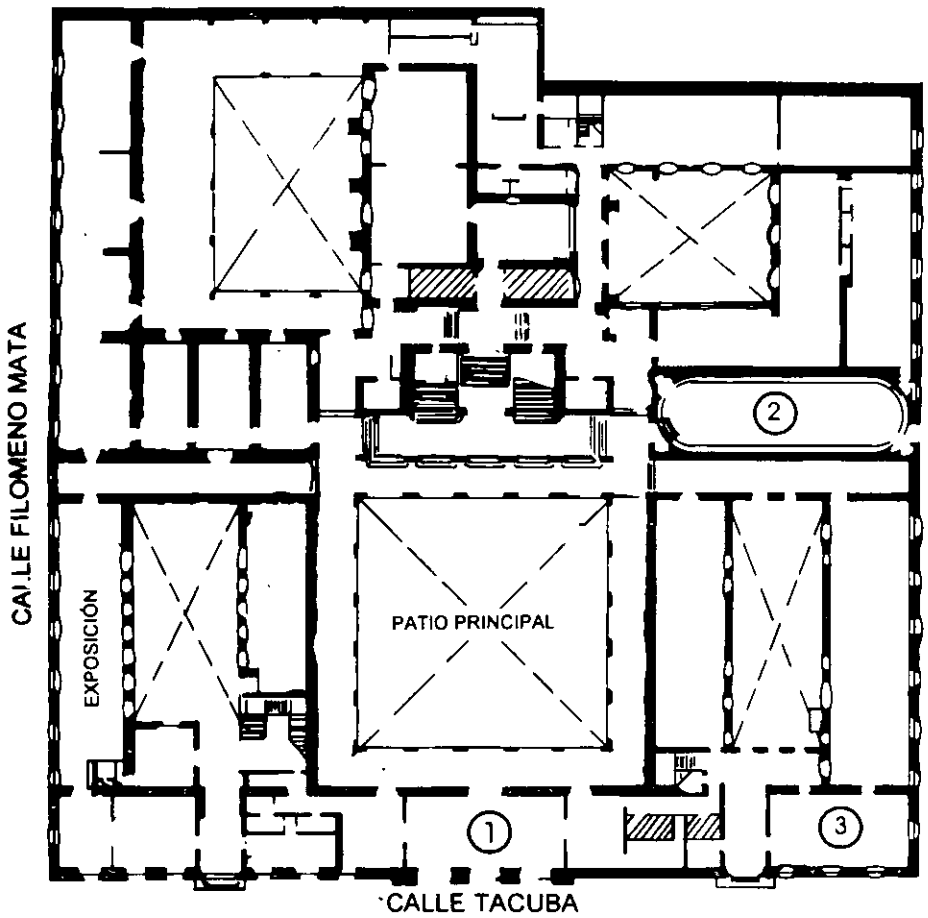
Con el objeto de mejorar los servicios que la División de Educación Continua ofrece, al final del curso deberán entregar la evaluación a través de un cuestionario diseñado para emitir juicios anónimos.

Se recomienda llenar dicha evaluación conforme los profesores impartan sus clases, a efecto de no llenar en la última sesión las evaluaciones y con esto sean más fehacientes sus apreciaciones.

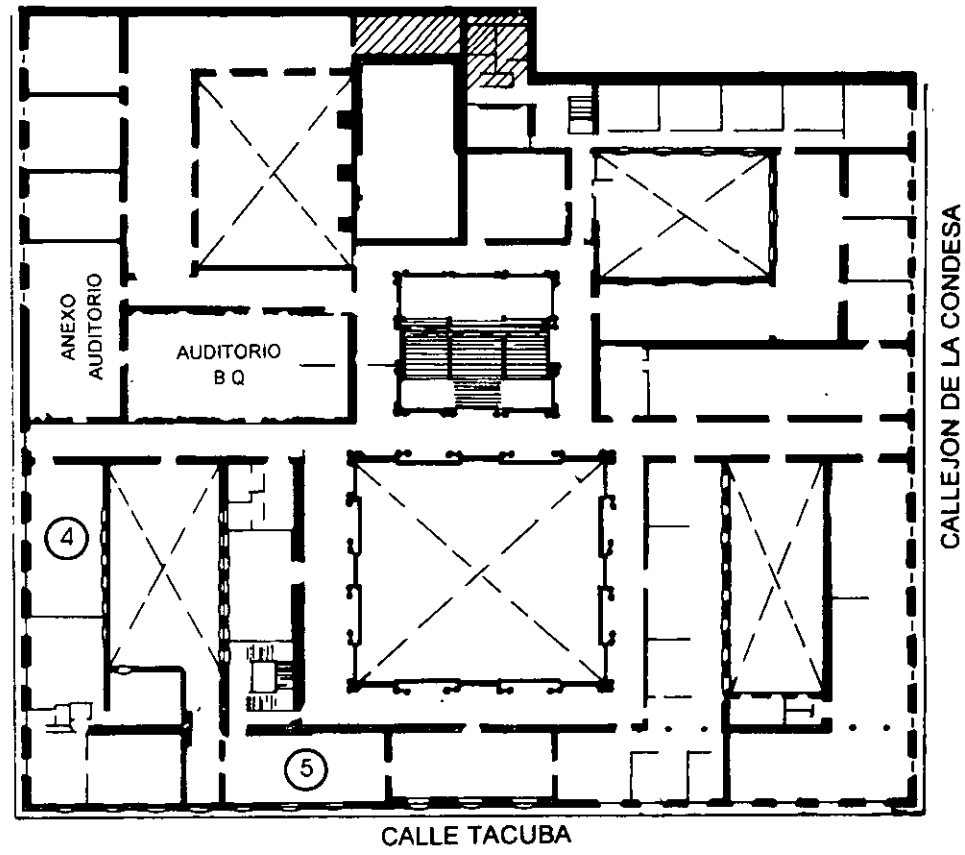
Atentamente
División de Educación Continua.

Palacio de Mineria Calle de Tacuba 5 Primer piso Deleg. Cuauhtémoc 06000 México, D F. APDO. Postal M-2285
Teléfonos: 512-8955 512-5121 521-7335 521-1987 Fax 510-0573 521-4020 AL 26

PALACIO DE MINERIA

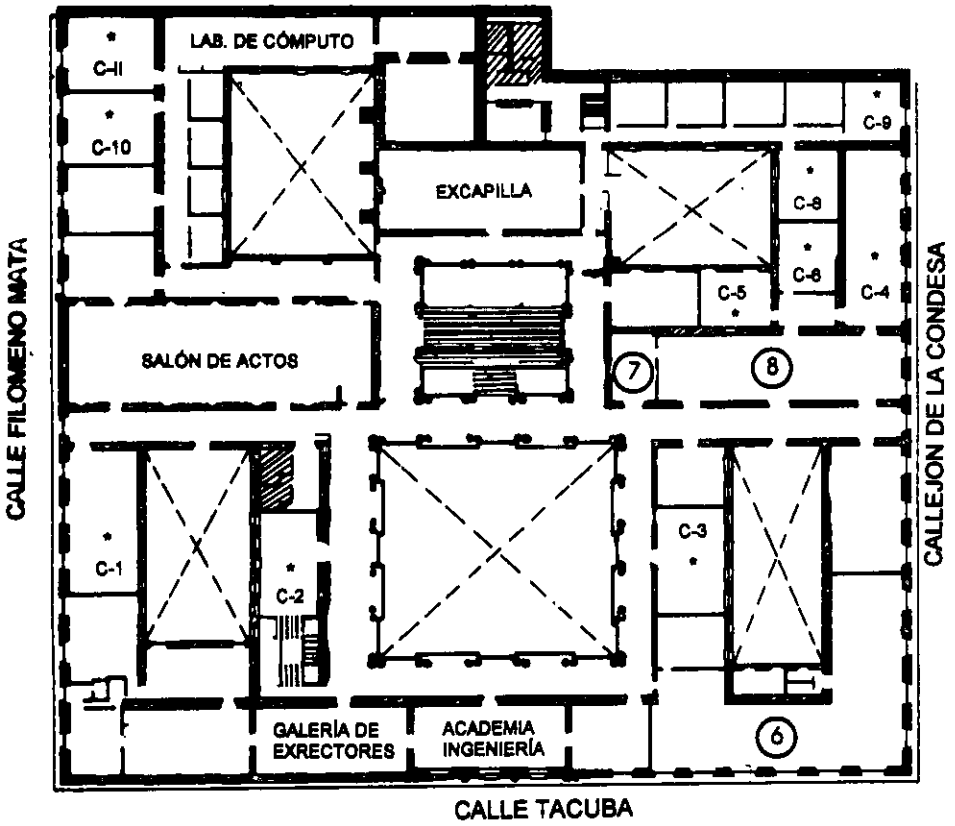


PLANTA BAJA



MEZZANINNE

PALACIO DE MINERIA



1er. PISO

GUÍA DE LOCALIZACIÓN

1. ACCESO
2. BIBLIOTECA HISTÓRICA
3. LIBRERÍA UNAM
4. CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN "ING. BRUNO MASCANZONI"
5. PROGRAMA DE APOYO A LA TITULACIÓN
6. OFICINAS GENERALES
7. ENTREGA DE MATERIAL Y CONTROL DE ASISTENCIA
8. SALA DE DESCANSO

SANITARIOS

* AULAS

DIVISIÓN DE EDUCACIÓN CONTINUA
FACULTAD DE INGENIERÍA U.N.A.M.
CURSOS ABIERTOS

DIVISIÓN DE EDUCACIÓN CONTINUA
DEC

FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA

CURSOS INSTITUCIONALES

BANCO DE MEXICO

3 al 16 de noviembre de 1998

TEMA: **VISUAL FOXPRO 5.0**

Ing. Claudia C. Zavala Díaz
Palacio de Minería
1998.

MANUAL DE

VISUAL FOXPRO 5.0

Corporativo DASH S.A. de C.V. Ing. Claudia C. Zavala Díaz.

TEMARIO

1. Introducción al ambiente visual y su terminología
2. Creación y manipulación de bases de datos e índices
3. Introducción a las clases, objetos y eventos
4. Formas
5. Manipulación y programación de Clases y eventos
6. Las clases y Librerías
7. Reporteador
8. Las Consultas
9. Ejecutable

Capítulo I. Introducción

I.1 Ventajas de Visual FoxPro

Visual FoxPro es un potente manejador de bases de datos con un entorno para desarrollo de aplicaciones, proporciona herramientas necesarias para el manejo directo de los datos. Gracias a la potencia y velocidad de Visual FoxPro procesos que se realizaban en computadoras multiusuario ahora es posible realizarlas en una PC.
Enumerando a las herramientas de Visual FoxPro tenemos las siguientes:

Es posible utilizar el Generador de clases, lo cuál disminuye en gran medida el tiempo de programación.

Las herramientas de Cliente/servidor y las mejoras del lenguaje, la cuál permite acceder a bases de datos remotas y a datos almacenados en un formato Xbase. Es posible crear en la PC un prototipo de herramienta Cliente/servidor y posteriormente transportar la aplicación final a un entorno Cliente/servidor sin tener que cambiar la programación.

Posee herramientas visuales que le permiten al usuario crear con facilidad aplicaciones tales como: informes, pantallas, etiquetas, etc. Esto reduce en gran medida el tiempo de programación.

El diccionario de datos le permitirá restringir y validar información desde la captura. Es posible utilizar nombres de campos mas grandes (hasta 128 caracteres), establecer relaciones entre tablas y utilizar el Generador de Integridad referencial para reforzar la validación entre datos.

Los asistentes y generadores los cuáles, guiarán al usuario a través de este para crear, con base en plantillas establecidas aplicaciones.

Los sistemas multiusuario requieren poco código adicional. Visual puede establecer diferentes medios de almacenamiento, los cuáles protegen la integridad de la información.

I.2. ¿Qué es la programación orientada a objetos?

La programación orientada a objetos para crear aplicaciones no es un concepto difícil de comprender, ***simplemente es una nueva forma de pensar en***

programación. Implica una nueva sintaxis la cuál es fácil de aprender, realmente la dificultad consiste en comprender la idea de ***aislar operaciones dentro de objetos.***

I.3. Definición de términos

Dentro de la programación orientada a objetos tenemos una nueva terminología, la que nos ayudará a comprender con mayor facilidad el manejo de la programación.

Clases y subclases: Son plantillas para los objetos, las cuáles se utilizarán para crear patrones de programación aplicables a cualquier aplicación.

Métodos: Son procedimientos vinculados a objetos y se ejecutarán como respuesta a una orden o evento.

Objetos: Se encuentran formados por clases que combinan datos y propiedades.

Propiedades: Son las características o atributos de un objeto o clase.

I.4 ¿Qué es un objeto?

Un objeto es una clase basada en sí misma o una colección de clases que realizan una función específica.
Los objetos pueden tener propiedades y métodos en común con otros objetos. Por ejemplo: un objeto es un botón donde puede tener las propiedades de color, tamaño, etc.

I.5 Funcionamiento de los objetos.

Los objetos tienen un conjunto de eventos asociados a ellos (cualquier cosa que le suceda al objeto). Por ejemplo un click del mouse, un enter, etc. Cuando un evento ocurre, éste dispara la ejecución de métodos con el mismo nombre que el evento.

Los métodos son procedimientos o programas vinculados al objeto y solicitados por los eventos asociados al objeto.

I.6 La programación orientada objetos en Visual FoxPro

Visual FoxPro tiene la posibilidad de crear objetos y clases utilizando el lenguaje. Los generadores nos ayudarán a crear con mayor facilidad y rapidez aplicaciones, pudiendo utilizar éstos posteriormente como clases. Es posible establecerle las propiedades a un objeto mediante código o por medio del generador. Aquí el tamaño del código dependerá del programador.

I.7 Elementos de Visual

Dentro de las partes que conforman un proyecto en Visual, al igual que en las versiones anteriores, tendremos los siguientes elementos:

- Proyecto: Conjunto de objetos que constituyen una aplicación.
- Formas: Cualquier tipo de pantalla, pudiendo ser de consulta, captura, etc.
- Informes: Reportes
- Etiquetas: Reporteador especial para etiquetas
- Querys: Relaciones entre tablas con generación de reportes, tablas, formas.
- Programas: Código externo que se utilizará para correr alguna aplicación llamada en una forma.
- Archivos de imágenes: Cualquier imagen que se utilice en la aplicación. Pudiendo ser en formato ico y bmp.

Capítulo II.
Creación y manipulación de bases de datos e índices

II.1 ¿Qué es una base de datos?

Base de datos
A diferencia de en las versiones anteriores de FoxPro donde una base de datos era un conjunto de datos. A partir de la versión en Visual una base de datos **es un conjunto de tablas, índices y relaciones.**

Tabla
Las tablas son las que contienen la información o datos.

Relaciones
Es la relación o vínculo que existen entre las tablas dentro de la base de datos. Las relaciones se harán mediante los datos llave de cada tabla.

Por lo anterior podemos decir que las bases de datos es el universo de la información dividida de acuerdo a su naturaleza, incluyendo sus índices y la relación que existe entre ellas.

II.2 Creación de bases de datos

Antes de crear cualquier base de datos para una aplicación en especial será necesario tomar en cuenta lo siguiente:

1. Conocer el problema.
2. Establecer cuáles serán las entradas y salidas de la información.
3. Datos que alimentarán al sistema y datos que arrojará así como los formatos de salida.
4. Analizar los procesos que se realizarán con la información.

Una vez que se han cumplido estos cuatro requisitos se realizará un análisis mas profundo de la información. Dividiendo los datos en tablas, mientras mayor sea el número de tablas será mejor, ya que esto nos indicará que en cada tabla se encontrarán sólo datos que tiene relación entre si y que se pueden interrelacionar con otras tablas. (este es el concepto básico de bases de datos relacionales)
Por ejemplo: Si nuestra aplicación es la venta de bienes, tendríamos lo siguiente:

- tabla de almacén o inventario

- tabla de registro de ventas
- tabla de registro de compras
- tabla con los datos de los clientes
- tabla con los datos de los distribuidores

Esto nos permitirá analizar la información por separado y su funcionamiento al relacionarlas.
Una vez que se ha dividido los datos de acuerdo a su naturaleza se deberá de hacer lo siguiente:

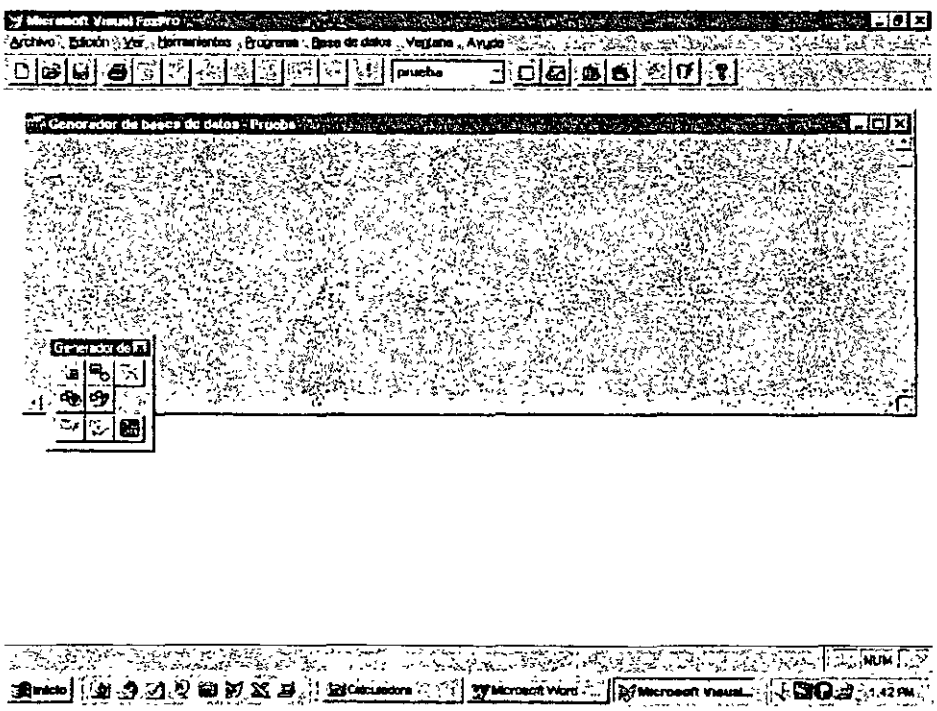
- Definir el nombre de los campos, su naturaleza (tipo de dato) y longitud
- Establecer llaves primarias y secundarias.

Ya que realizó esta recopilación y análisis previo ahora se procederá a crear la base de datos, sus tablas e índices.

Creación de la base de datos

Para ello llamaremos a la opción del menú Archivo-Nuevo-Base de datos
Aparecerá una pantalla donde se nos solicitará el nombre de la base, éste deberá de ser un nombre inteligente, esto es un nombre que por sí sólo nos indique el tipo de información que almacena.

Hecho esto nos aparecerá la siguiente pantalla, en donde definiremos las tablas que compondrán a la base de datos.



Se nos muestra una ventana en gris vacía y una barra de herramientas.
Este es el concepto visual de una base de datos. Ahora será necesario definirle las tablas.

Creación de tablas

Utilizaremos la barra de herramientas, para el caso de crear una nueva tabla. Se deberá de oprimir el botón con el asterísco o es posible utilizar el botón derecho del mouse y seleccionar Crear tabla.
Visual nos solicitará el nombre de la tabla. Recordemos que el nombre deberá de estar íntimamente ligado a la naturaleza de la información.

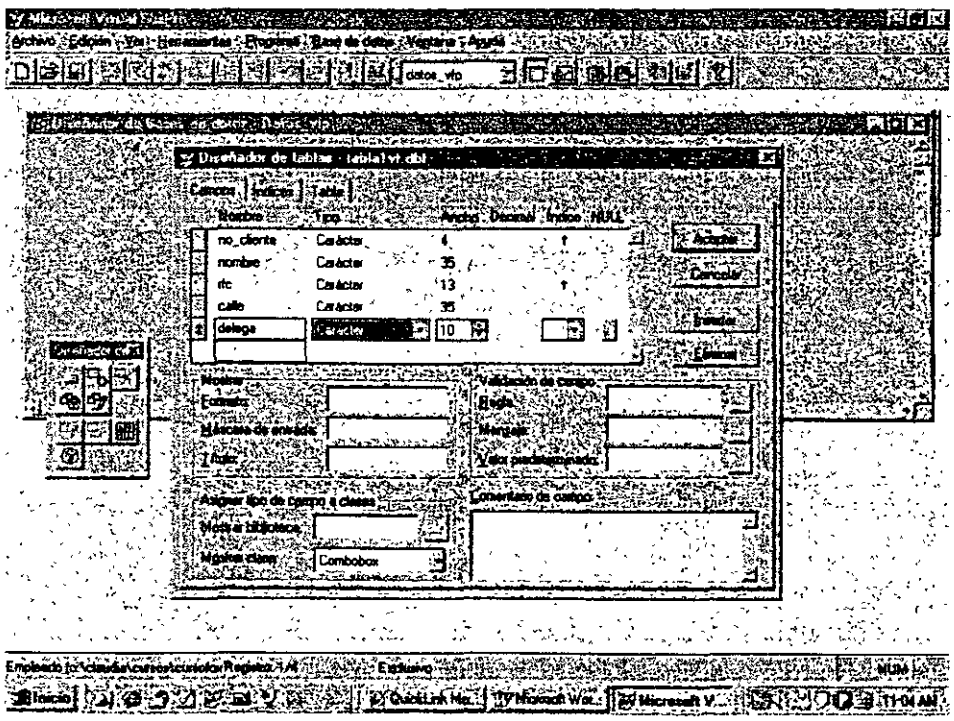
Analizando la pantalla tenemos lo siguiente:

Anexar tablas

Si la tabla ya fue creada y es necesario sólo agregarla, entonces se seleccionará agregar tabla o de la barra de herramientas el botón con el signo de mas.

Para el caso de que hubieran sido tablas creadas con anticipación fuera de una base de datos se añadirán o se irán creando conforme se va avanzando en el proyecto.

II.3 Creación de tablas



Para crear las tablas bastará con oprimir el botón del asterístico en la barra de herramientas u oprimir botón derecho del mouse y seleccionar Agregar tabla.
El sistema nos mostrará una pantalla ya conocida en las versiones anteriores.

En la parte superior se nos muestra un recuadro con el nombre de la tabla y a lado de ésta el nombre de la base de datos a la que pertenece, un botón donde nos señala que de quererlo o necesitarlo ahí podemos definirle propiedades específicas a la tabla en general.
Debajo encontramos tres pestañas la primera nos mostrará los campos en la tabla con sus propiedades y restricciones, la segunda los índices y la última pestaña con las características y propiedades de tabla.
Ahora en el área media definiremos el nombre de los campos, el tipo de dato, los decimales (en el caso de ser numéricos), si acepta o no valores nulos y si es o no campo índice.

II.3.1 Tipos de datos

Dentro de los tipos de datos en Visual tenemos:

Tipo	Descripción	Longitud	Restricciones
Carácter	Cualquier texto, inclusive numéricos no operables	De 1 hasta 254	
Moneda	Importes monetarios. Para el caso de mas de 4 decimales los redondeará a cuatro	8 bytes	-922337203685477.5808 hasta 922337203685477.5807
Fecha	Datos cronológicos que consta de dia, mes y año	8 bytes	01/01/100 hasta 31/12/9999
FechaHora	Datos cronológicos que consta de dia, mes año y hora	8 bytes	01/01/100 hasta 31/12/9999, y 00:00:00 A.M. hasta 11:59:59 p.m.
Lógico	Valor booleano	1 byte	Verdadero(.T.) o Falso(.F.)
Numérico	Enteros o fracciones	8 bytes	-,9999999999E+19 hasta ,9999999999E+20
Doble	Número de signo flotante de doble precisión	8 bytes	+/-4,94065645841247E-324 hasta +/-1,79769313486232E308
Flotante	Igual que numérico		
General	Referencia a un objeto OLE(imagen, sonido, etc.)	4 bytes	Limitado por la memoria de la computadora
Entero	Valores numéricos sin decimales	4 bytes	-2147483647hasta2147483646
Memo	Referencia a un bloque de datos	4 bytes	Limitado por la memoria de la computadora
Carácter(binario)	Cualquier dato de caracteres que desee mantener sin cambiar la tabla de códigos	De 1 a 254	
Memo (binario)	Cualquier dato de campo memo que desee mantener sin cambiar de tabla de códigos	4 bytes	Limitado por la memoria de la computadora

II.3.2. Propiedades del campo

Definiremos como propiedades del campo a aquellas que definen el comportamiento natural del campo, esto es validaciones, valores por default, textos, etc. En esta versión encontramos divididas a las propiedades: Mostrar, Validación del campo y asignar tipo de campo a clases.

Mostrar:

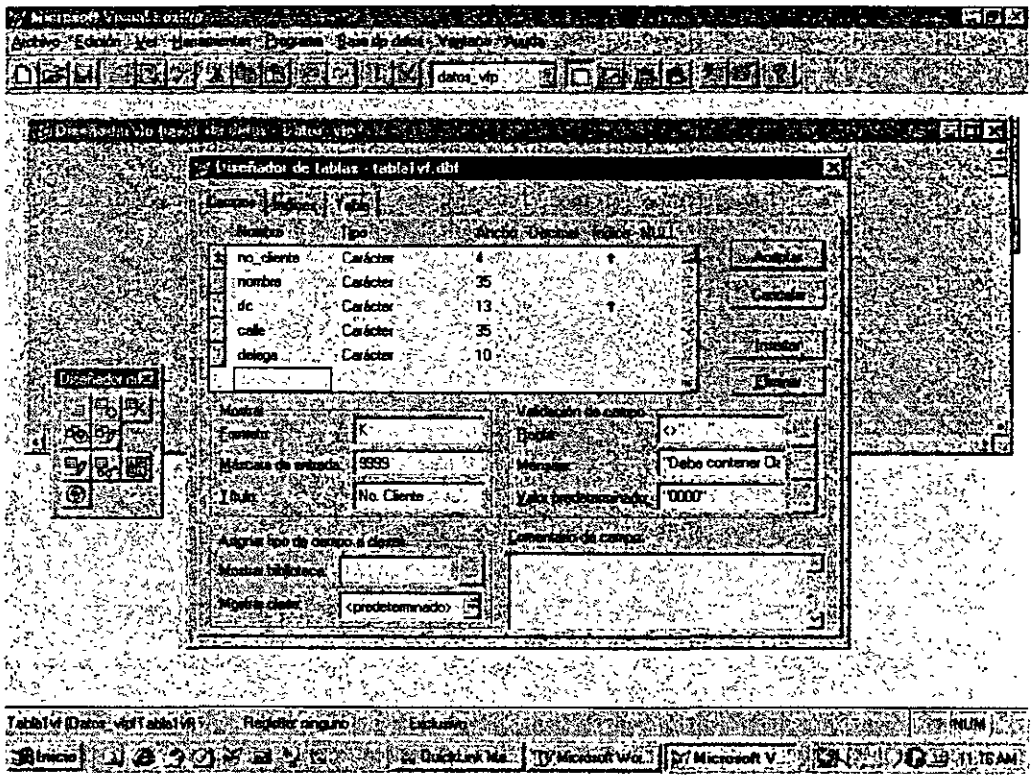
- **Formato**: Formato de entrada de la información, por ejemplo: mayúsculas, que marque la información para ser eliminada, etc.
- **Máscara de entrada**: Cómo permitirá la entrada de información: Sólo letras, letras y números, etc.
- **Título**: Título del encabezado de columna en el despliegue de la información

Validación del campo:

- **Regla de validación**: Es donde le podemos definir la validación que manejará el campo. Por ejemplo que el dato no sea posible dejarlo en blanco.
- **Mensaje**: Es el texto que mostrará Visual en el momento que la regla de validación se trate de violar.
- **Valor** : Es el valor que por default nos mostrará el campo.

Asignar tipo de campo a clases

- **Librerías de clases**: Librería donde se encuentra la clase asociada al campo.
- **Clase**: Tipo de control que aparece por defecto al incluir el campo en algún formulario.



II.3.3 Índices

Los índices son aquellos datos que servirán de enlace con otras tablas y ayudará en la búsqueda de información

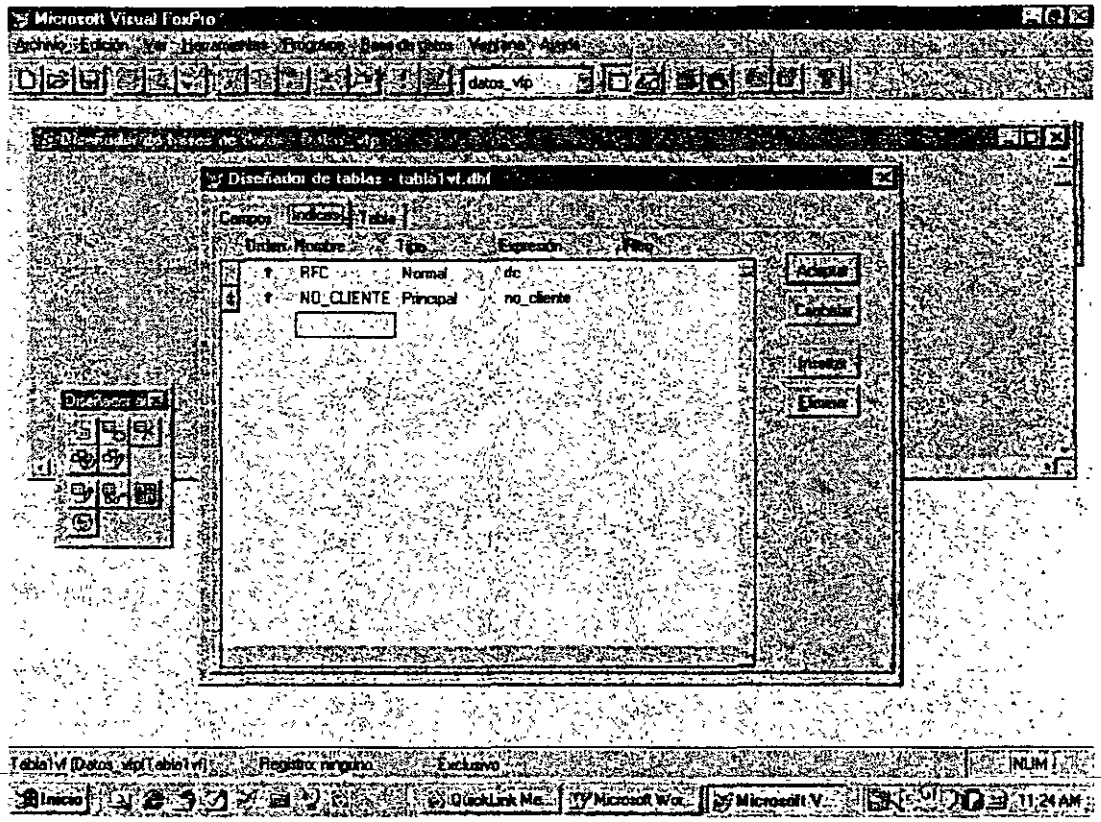
Existe tres diferentes tipos de índices:

- **Principal**: Es aquel dato único que se utilizará para diferenciar un registro de otro. Por ejemplo. La clave de cada producto es única, la existencia puede ser mucha pero el producto único.

- **Regular**: Son llaves secundarias que sólo se utilizarán para ordenar de diferentes formas la información contenida en una tabla.

- **Único**: Es el índice que no mostrará mas que la primera ocurrencia de cada dato. Por ejemplo. No. de cliente en la tabla de ventas, aún cuando el cliente hubiera comprado varias veces su registro sólo aparecerá una sola vez.

- **Candidato**: Es aquel índice que podría llegar a ser principal sin serlo. Es otra alternativa para datos que no se repiten pero que no son únicos.



Analizando los datos que solicita la pantalla se encuentran:

- **Nombre**: Que se le asignará al índice.
- **Tipo**: Tipo de índice. Se seleccionará uno de la lista.
- **Expresión**: Es el o los campos por medio de los cuales se ordenará la información. Aquí debemos recordar que si el índice lo formarán mas de un campo será necesario que ambos sean del mismo tipo.
- **Filtro**: Este se utiliza cuando se requiere que el índice sólo considere a cierta información. Por ejemplo. Todos los que su no. de cliente es menor a "200".

Dentro de la definición de los índices se crearán todos aquellos que sean necesarios para nuestra aplicación. Todos serán almacenados en un solo archivo llamado igual que la tabla y con la extensión CDX. Para diferenciar un índice de otro se hará mediante la selección de TAGS.
Al abrir cualquier tabla que contenga índices, éstos se abrirán simultáneamente, pero ninguno estará activo.

Una vez que se han creado los campos necesarios y los índices correspondientes, se oprimirá el botón de ACEPTAR.

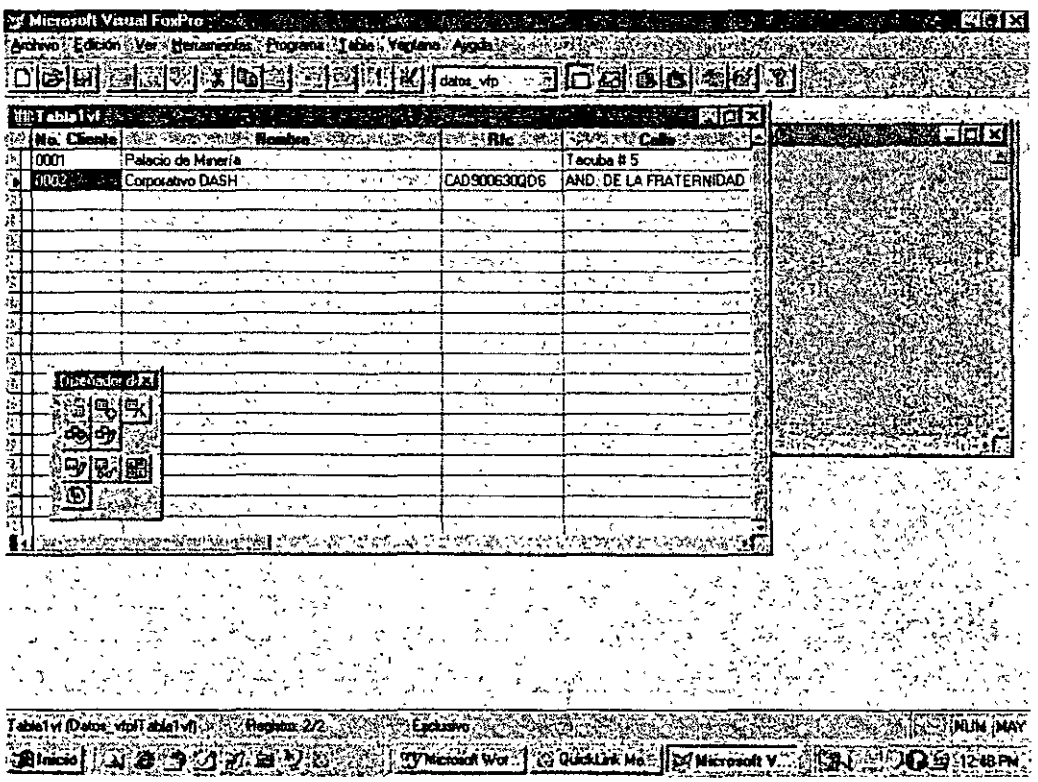
II.4 Desencadenantes

Es código que se ejecuta cuando se produce una inserción, modificación o eliminado de registros dentro de la tabla.

II.5. Tablas

II.5.1 Modificar, visualizar y añadir información

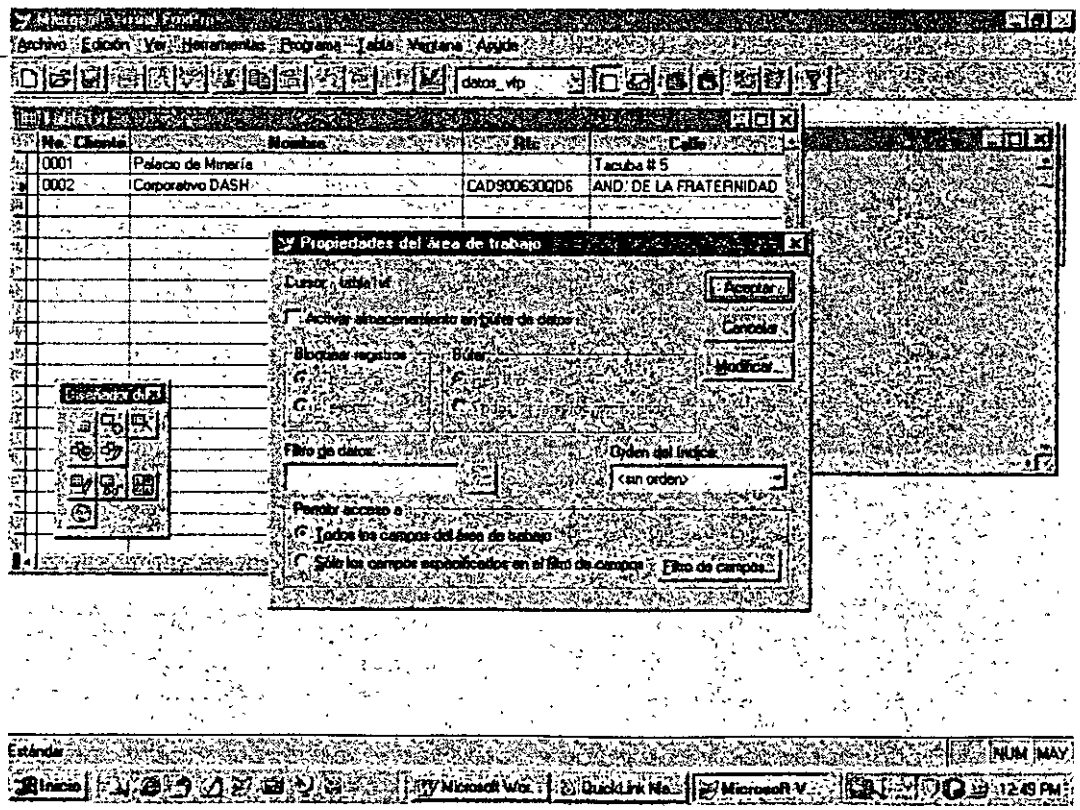
Para visualizar y modificar información lo podremos hacer teniendo la base de datos abierta, se marca a la tabla y mediante el botón derecho del mouse le diremos EXAMINAR, esto nos enviará a pantalla el contenido de la tabla.
Si lo hiciéramos mediante la ventana de comandos sería necesario abrir la tabla con USE TABLA y después especificarle que la abra para visualizar el contenido con BROWSE.



Para el caso de añadir información se hará de la misma manera pero además deberemos decirle a Visual que vamos a añadir, por lo que llamamos al menú VER-MODO AÑADIR. Notaremos que si definimos valores por default estos aparecerán al crear un nuevo campo.

II.4.2 Modificación de la estructura

Cuando después de capturar el primer campo surgen modificaciones a la estructura. Para ello llamaremos al menú del botón derecho del mouse y MODIFICAR, el sistema nos mostrará la misma pantalla que la mostrada cuando creamos por primera vez la tabla.
Otra forma de hacerlo es, si estamos en la vista de la tabla, esto es dentro de la información y llamamos al menú TABLA-PROPIEDADES. Se nos mostrará la siguiente pantalla, en donde podremos desde cambiar el orden del índice y las propiedades de la tabla.



En esta nueva pantalla encontraremos dos partes importantes: Definir algún filtro a la tabla y establecer el índice con el que va a trabajar. Si accesamos a la lista de Orden del índice veremos que nos muestra los índices existentes en la tabla.

Originalmente aparecerá que la tabla no tiene orden establecido esto es que, aún cuando existen índices no se ha especificado uno como maestro.

Si además de indicar cuál es el maestro se desea modificar la estructura se deberá de oprimir el botón MODIFICAR y nuevamente Visual mostrará la pantalla donde se definen los campos de la tabla.

II.4.3. Eliminación de registros

Para eliminar registros será necesario marcar por medio del mouse el registro deseado. La marca se deberá de hacer en la columna más pequeña encontrada en la parte izquierda de la vista de la tabla. Cuando es marcado se verá en color negro este recuadro lo que indica que está marcado para ser borrado mas no está eliminado.

Si se desean eliminar los registros marcados para borrar se deberá de teclear en la ventana de comandos la instrucción **PACK**, la cuál eliminará los registros que se encontraban marcados para ser borrados.

En el caso de que antes de borrarlas se deseara recuperar o eliminar las marcas de borrado de la tabla se deberá de teclear la instrucción **RECALL,** ésta eliminará las marcas de borrado. Para el caso de que ya se halla realizado el PACK no será posible recuperar ningún registro eliminado.

Para el caso de omitir los registros marcados para ser borrados para algún cálculo o visualización, se utilizará la instrucción **SET DELETE ON**.

Cuando lo que se requiere es eliminar a todos los registros de la tabla se hará mediante la instrucción **ZAP**, una vez utilizada ésta tampoco se nos permitirá recuperar la información.

CAPÍTULO III.
Introducción a las clases, objetos y eventos

En este capítulo se retomarán los conceptos vistos en el primer capítulo, para comprender de una manera más sencilla los conceptos que conforman la base de la programación orientada a objetos.

Clase

Una clase es una plantilla que define las características de un objeto y describe qué apariencia y comportamiento deberá tener.
Existe la Clase de base la cuál es una clase definida internamente por Visual FoxPro que puede utilizarse como base para otras clases definidas por el usuario. Por ejemplo, los formularios y todos los controles de Visual FoxPro son clases de base que puede ampliar con su propia funcionalidad para crear nuevas clases.

Instancia

Término utilizado en programación orientada a objetos. Un objeto creado a partir de una definición de clase. A diferencia de una clase, que solo es una definición, una instancia existe realmente como objeto que se puede utilizar para realizar tareas. Por ejemplo, un cuadro de texto de un formulario es una instancia de la clase TextBox.

Objeto

Un objeto es una instancia de una clase. Una clase es una descripción de los datos y las características de comportamiento de un grupo de objetos. Los objetos de Visual FoxPro pueden ser formularios, conjuntos de formularios o controles. Los objetos poseerán propiedades derivadas de la clase a la que pertenecen. Utilice objetos de Visual FoxPro para implantar un comportamiento coherente y fiable por toda la aplicación, para reducir la cantidad de código y para aumentar la reusabilidad del código.

Propiedad

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 estar visible. Las propiedades pueden modificarse mediante código o por medio de la ventana de propiedades.

Método

Una acción que un objeto es capaz de realizar.

Evento

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.

Bibliotecas de clases

Es el conjunto de clases almacenadas en un archivo. Todas las clases de Visual se encuentran almacenadas en una biblioteca. Es posible y recomendable que el usuario cree sus propias bibliotecas con clases nuevas que vaya creando.

Para comprender mejor lo anterior veremos el siguiente ejemplo:

La clase es la raza canina
Un objeto es un Pastor Alemán
Algún método realizado por el objeto sería el dormir o estar despierto, el color, tamaño, etc.
Y algún evento podría ser mover la cola al ver la comida. Él responde a un estímulo como es la comida, este proceso ya se encuentra grabado en un reflejo condicionado, que hará que se dispare al recibir el estímulo.

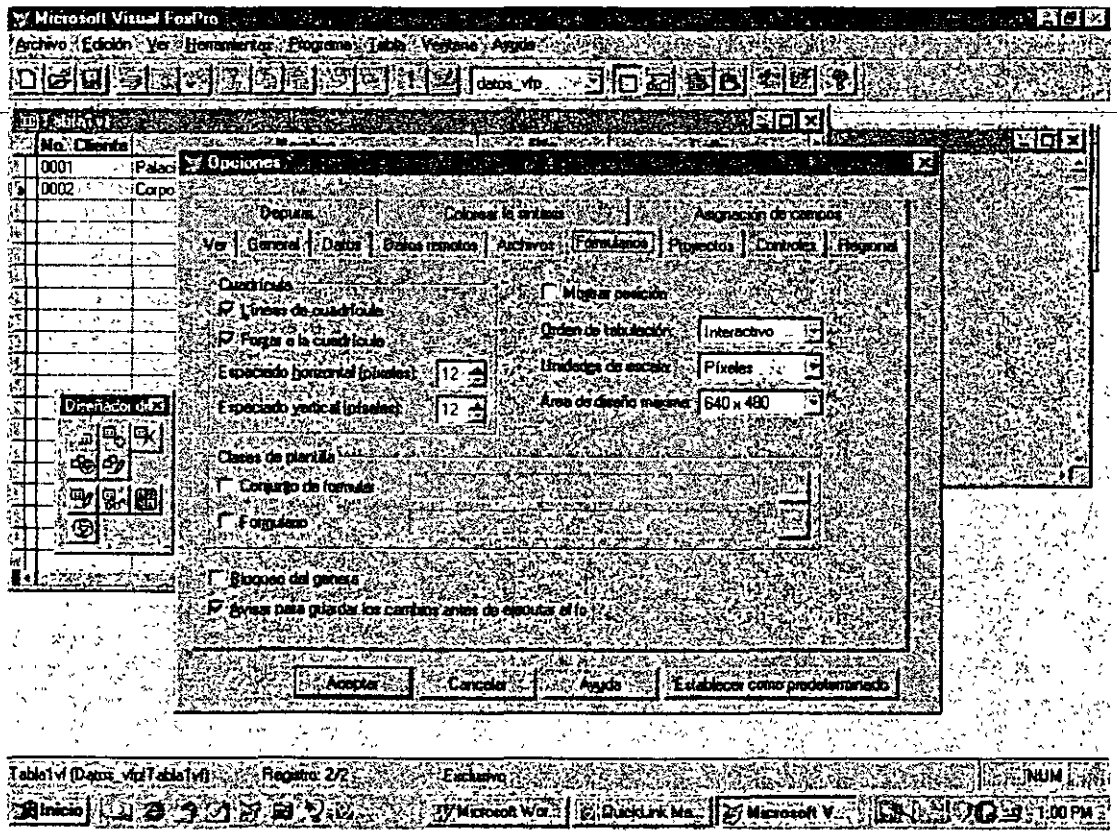
CAPÍTULO IV. Las Formas

IV.1. Definición

Las formas son pantallas que diseñará el programador para mostrar, capturar, modificar, etc. la información que tiene o tendrá la tabla. Anteriormente era común encontrar que se tuvieran pantallas para cada aplicación, veremos que ahora ayudado de las herramientas que nos proporciona Visual será mas sencillo y podremos tener todos los procesos en una misma pantalla.

IV.2 Opciones de configuración

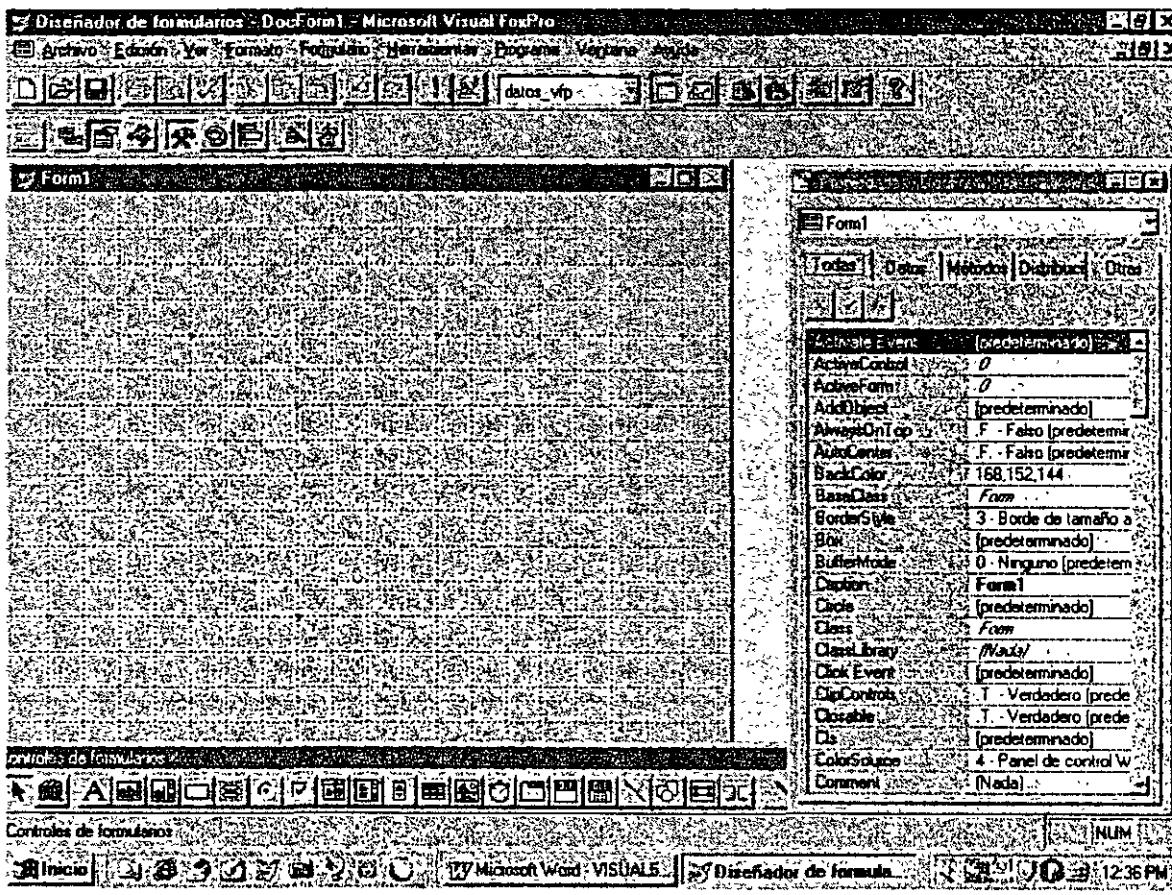
Antes de crear algun formulario es conveniente establecer las condiciones de trabajo, esto lo haremos mediante la opción de Herramientas y Opciones, mostrándonos la siguiente pantalla:



Es particularmente importante establecer el tamaño de la pantalla sobre la cuál se trabajará, esto es el área máxima de diseño: monitores con resolución de 640x480, 800x600, 1024x768 y 1280x1024.
La definición de esto mostrará al usuario el área que posee para diseñar.

IV.3 Creación

Llamaremos al menú ARCHIVO-NUEVO-FORMULARIO se abrirá una nueva pantalla semejante a la que se muestra a continuación.



Denotaremos en esta dos grandes partes una ventana cuadriculada (cuerpo de la forma) y la ventana de propiedades.
En la parte superior e inferior de la pantalla (en este caso) aparecen unas barras de herramientas, la superior hará llamadas a las distintas partes que conforman una forma y la inferior nos ayudarán al diseño.

IV.4. Ventana de Propiedades

La ventana de propiedades nos proporcionará todas las propiedades posibles de modificar dependiendo del objeto de que se trata, esto es cada objeto posee sus propiedades definidas.

Analizando las partes de la ventana de propiedades tenemos lo siguiente:

Objeto: Es una ventana que nos indicará el objeto del que muestra las propiedades. Pudiendo cambiarlo a través de la selección directa o de esta ventana.

Veremos que divide a las propiedades por su naturaleza.

Todo: Muestra todas las propiedades ordenadas alfabéticamente.

Datos: Todas las propiedades que tengan que ver con el tipo de dato a guardar en cada campo.

Métodos: Métodos existentes de cada objeto. Estos también varían de acuerdo al objeto. Aquí es donde definiremos los procesos a realizarse dependiendo del método utilizado.

Distribución: Todas las propiedades que definirán la presentación del objeto. Se puede decir que son los adjetivos calificativos de éste. Por ejemplo, color tamaño, tipo de letra, etc.

Otros: Son otras propiedades que no pertenecen a ninguna de las clasificaciones anteriores.

Cabe aclarar que cada propiedad posee valores específicos que se pueden manejar, nos mostrará valores por default en algunas propiedades, en otras el valor es más libre.

IV.5 Entorno de datos

Generalmente cuando se crea una nueva forma de captura el objetivo principal es manipular información de una o varias tablas. Por lo que el primer paso en el diseño de una forma será definirle las tablas a utilizar.

El Entorno de datos será la inclusión de las tablas a las que se hace referencia en la forma.
Es importante mencionar que esto nos ahorrará el conocido y molesto USE y SELECT, debido a que Visual asignará un área de trabajo específica a cada tabla.

Para llamar al entorno de datos se hará lo siguiente: Por medio del botón derecho del mouse y Entorno de datos o por medio de la barra superior de herramientas el botón que tiene como una pantalla y ligada a ella otras dos.

Dentro del entorno de datos se mostrarán también las relaciones entre tablas.
En la siguiente figura veremos que cada tabla tendrá su menú de propiedades también en donde destacaremos las principales.

Alias: Sobre nombre por medio del cuál hará referencia el compilador a la tabla. Por ejemplo, anteriormente utilizábamos la siguiente sintaxis

SELECT 1
USE BASE

Donde el número 1 sería el alias o la etiqueta por medio de la cuál denominaremos a BASE

En este caso Visual sobrenombra a la base con su mismo nombre por lo que a través de las diferentes rutinas que generemos sólo será necesario hacer referencia al alias.

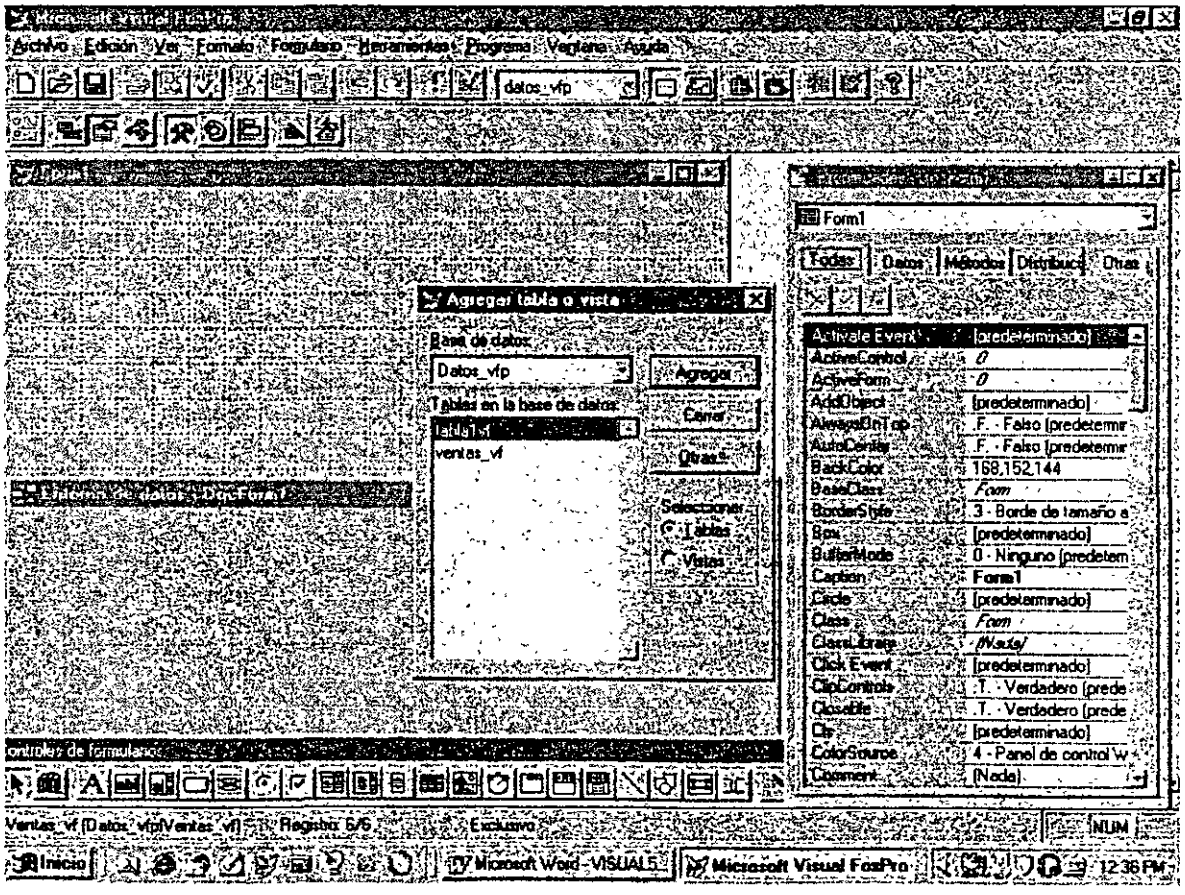
SELECT BASE

Database: Base de datos a la cuál pertenece la tabla. Este dato es de sólo lectura.

Exclusive: Aquí le definiremos si la tabla es de uso exclusivo o compartido, esto nos servirá para el manejo de tablas en red.

Order : Índice maestro con el que trabajará la tabla.

ReadOnly: Si la tabla es de sólo lectura o también se podrá escribir.



IV.6 Objetos

Se verán a continuación cada uno de los objetos que es posible incluir en una forma.

Label (A): El objeto texto es cualquier etiqueta o letrero que incluyamos dentro de la forma. Visual los irá membretando secuencialmente, esto sucederá con todos los objetos. Y los llamará *LABEL*. Esto es, LABEL1, LABEL2... así hasta el último incluido.

Cuadro de texto (ab): Lugar donde se almacenará información. Asemejándolo con versiones anteriores diremos que es una variable donde se solicitará información GET. Visual lo llamará *TEXT*.

Cuadro de edición (al): Es un lugar donde se podrá capturar o mostrar información, esta se encontrará dentro de una ventana con desplazamiento. Es por esto que, se le denomina región de edición. Visual le denominará *EDIT*.

Botón de comando: Es un solo botón al cuál se le asignará código para realizar operaciones específicas. Dentro de Visual será *COMMAND*.

Grupo de comandos: Es un objeto que reúne a varios botones comando, este caso tendrán comportamiento como grupo y en forma individual. Cualquier botón que sea seleccionado accionará a un proceso o llamará a otra aplicación. Son objetos que se utilizan como órdenes directas al sistema. Se llamará *COMMANDGROUP* en su forma de conjunto e individual siguen siendo *COMMAND*.

¿De que manera diferenciará Visual al command1 del command1 que pertenece a un grupo?. Dentro de la pregunta se encuentra la respuesta Visual hará referencia directa a la inclusión de un objeto dentro de otro. Esto es, el botón aislado lo llamará simplemente COMMAND1, pero al que pertenece al grupo le llamará primero por su dominio en este caso el COMMANDGROUP y después al objeto COMMAND. La notación sería la siguiente:
COMMANDGROUP1.COMMAND1. Si esto se leyera de derecha a izquierda leeríamos lo siguiente: El botón de comando X que pertenece al grupo de comando X.
La notación se escribirá de izquierda a derecha de lo más a lo menos general.

Grupo de opciones: En este caso tendremos en lugar de botones de oprimir, radio botones. Al igual que el grupo de comandos el grupo de opciones estará formado por un conjunto de éstas. Se llaman *OPTIONGROUP* y las opciones individuales *OPTION* .
En este caso el manejo del objeto es diferente ya que las opciones son mutuamente excluyentes, esto es no es posible seleccionar mas de una a la vez. Por ejemplo, una persona no puede ser sindicalizada y de confianza al mismo tiempo, por lo que sólo se podrá elegir una de las opciones. Si una es seleccionada automáticamente deseleccionará a la otra.

Casilla de verificación: En este caso es un objeto individual el cuál se seleccionará o no, tan solo posee dos estados 1 o 0. En el momento de ser seleccionado aparecerá una cruz dentro de él. Visual lo llama *CHECK*.

Cuadro combinado: Lista de opciones en donde se podrá elegir la deseada. La lista aparecerá enrollada, bastará con oprimir la flecha con el mouse para poder desplegar el contenido total de ésta. Lo llamaremos *COMBO*. Este objeto se utilizará para cuando se tienen opciones fijas de selección y no se desea que el usuario teclee la información como quiera, esto nos ayudará a tener un control más exacto de los datos. Por ejemplo, la delegación, el estado, el país, departamento, etc.

Cuadro de lista: La diferencia con la lista anterior es que esta se desplegará en un área mayor dentro de la pantalla pudiéndonos desplazar mediante las flechas de navegación a través del contenido de la lista. En este caso es un objeto que podremos rellenar con campos de una tabla, con el contenido de un campo dentro de una tabla, etc. Se denomina *LIST*.

Control numérico: Es el objeto que se utiliza para manejar valores numéricos. Por ejemplo la antigüedad de un empleado dentro de una empresa. Simplemente aumentará o disminuirá el usuario el número mediante las flechas que posee la caja en la que se encuentran los valores. Visual lo denomina *SPINNER*.

Cuadrícula: Es un objeto con características muy especiales. Se utiliza para desplegar el contenido de una tabla es muy semejante a un browse o despliegue por sistema de la tabla, con la diferencia que aquí podremos hacer validaciones de los campos, incluir y excluirlos, trabajarlos como entes por separado.
Una cuadrícula está compuesta por tres elementos: La columna en general (*COLUMN*), el encabezado de la columna (*HEADER*) y el dato a mostrar o almacenar (*TEXT*).
Es un objeto muy útil en cuanto despliegue de información se trata. La cuadrícula es llamada GRID.

Timer: Objeto que hace que un proceso se esté generando constantemente. Lo llamaremos *TIMER*.

Marco de página: Cuando poseemos demasiada información que se desea aparezca en una pantalla, es conveniente dividirla en forma razonable en una o dos pantallas. El marco de página nos permite hacer esto sin que cada pantalla sea una forma nueva. Dentro de este podremos manejar folders diferentes con información distinta. Por ejemplo, se manejarán datos de clientes, el primer folder serían los datos generales, en el segundo las compras y en el tercero las condiciones de pago.
Nuevamente tenemos que el objeto completo es el marco y se encuentra formado por páginas y cada página posee objetos.
Al marco llamaremos *PAGEFRAME*, a las páginas *PAGE* y los objetos de acuerdo a su naturaleza.
Recordando un poco lo visto de las jerarquías como llamaríamos a un botón de comandos dentro de una página en una cuadrícula.

PAGEFRAME1.PAGE1.COMMAND1

Donde al igual que en el ejemplo anterior primero es el objeto de mayor jerarquía, así hasta llegar al de menos.

IV.6.2 Propiedades generales

Llamaremos propiedades generales a todas aquellas que se utilizan en todos o casi todos los objetos que podemos incluir en una forma.

Propiedades de datos

ControlSource: Será el lugar donde le determinemos al objeto como se llamará la variable que almacenará el dato capturado por éste o el campo de la tabla de donde se obtendrá la información.

Mas adelante veremos específicamente por objeto las propiedades de datos restantes.

Propiedades de distribución

Alignment: Alineación del texto respecto al área definida por este.

Backcolor: Color que se utilizará como fondo en los objetos. Será una tercia de colores que nos dará la combinación deseada. Esto se asemeja al ColorRGB de la versión anterior. Al dar doble click en esta propiedad el sistema nos mostrará los colores que es posible seleccionar, al elegir uno Visual define la combinación correspondiente.

Caption: Título que se le asignará al objeto. Por ejemplo, en las formas será el nombre que aparecerá en el marco.

Disablebackcolor: Color del que se verá el fondo del objeto cuando se encuentra deshabilitado.

Disableforecolor: Color del que se verá la letra o contenido del objeto cuando éste se encuentre deshabilitado.

Fontbold: Para el caso de que se deseara que la letra estuviera en bold o remarcada.

Fontname: Nombre del font o tipo de letra a utilizar. Por default utiliza la letra Arial.

Fontsize: El tamaño del font o tipo de letra. El default es de 10 puntos pero algunas veces se requiere que sea más grande en el caso de títulos o mas chico cuando son etiquetas de otros objetos.

Forecolor: Color de la letra dentro de un objeto.

Height: Alto del objeto.

Left: A que distancia se encuentra de la izquierda de la orilla de la forma el objeto.

SpecialEffect: Tipo de efecto especial. En este caso podría mostrarse como un objeto plano o con efecto de tercera dimensión. Por default utiliza este último. A veces es conveniente dejarlo en forma plana y crearle un efecto de sombra.

StatusBarText: Texto que desplegará en la barra de estado. Este texto se utilizará como texto de ayuda para el usuario, donde el programador le dará una breve explicación del tipo de dato que se almacenará o las restricciones utilizadas. Este texto se desplegará en el momento que el usuario se encuentre en el objeto que lo posee.

ToolTipText: Este texto es una muy breve definición del objeto. Este se mostrará tan solo al pasar el mouse sobre el objeto, no es necesario oprimirlo. Esta propiedad se encuentra íntimamente ligada a la Propiedad SHOWTIPS de la forma. Para el caso de que se han definido tooltips en algunos objetos, pero la propiedad showtips en la forma se encuentra en Falso, el sistema no mostrará las ayudas. Por lo que es necesario que la showtips se encuentre en verdadero.

Top: A que distancia del tope superior de la forma se encuentra el objeto. Esta propiedad es muy útil para alinear los objetos.

Visible: Propiedad que nos permitirá jugar con los objetos en cuanto a que sean o no vistos. Por ejemplo, se desea que un cierto objeto aparezca sólo cuando una condición resultó verdadera, por lo que en el diseño se incluye este objeto, pero se le da la propiedad de visible igual a falso y por medio de código se cambiará ésta a verdadero.
Es muy importante mencionar que en el diseño de la forma deberán de incluirse todos aquellos objetos que se utilicen en algún momento, aún cuando aparezcan y desaparezcan como resultado de algún proceso o condición.

Width: Ancho del objeto.

Otras Propiedades

Enabled: Esta propiedad nos permite habilitar o deshabilitar a un objeto. Por ejemplo, el botón de grabar los datos de captura se encuentra deshabilitado hasta que no se termina de capturar lo mas importante del registro. Esta propiedad también es posible modificarla mediante código.

Name: Nombre que se le asigna al objeto. Anteriormente mencionamos que Visual le da por default el nombre de acuerdo a la clase a la que pertenece el objeto adicionándole un número secuencial. Este nombre es posible modificarlo.

TabIndex: Es el orden en el que el tabulador irá viajando entre los objetos. Este número se le asignará al objeto de acuerdo al orden en el que se fueron anexando a la forma. Muchas veces este número será necesario modificarlo ya que se van anexando objetos conforme se avanza en el proyecto y esto nos hace perder la secuencia lógica.

IV.6.3 Propiedades específicas

Forma

- **Autocenter**: Cuando muestra la forma centrada respecto a la pantalla.
- **BorderStyle**: Tipo de línea que utilizará para el borde de la forma (*Ajustable*, sin borde, doble y sencillo)
- **Closable**: El que aparezca o no el botón de cerrar la forma. Esto es algo que deberá de definir el programador, ya que puede darse el caso de que el usuario cierre la forma sin que hubiera terminado de capturar.
- **ControlBox**: Si se desea que aparezca o no el botón del extremo superior izquierdo de la forma donde nos permitiría cambiarnos entre ventanas, cerrarla, etc.
- **Desktop**: La forma desktop es aquella que ocupa todo el espacio del escritorio de trabajo de Windows.
- **Icon**: Es el icono que deseamos que aparezca cuando la forma se encuentra minimizada.
- **LockScreen**: Candado que se le pone a la forma para que no sea visualiuzada.
- **Maxbutton**: El que aparezca o no el botón de maximizar.
- **Minbutton**: El que aparezca o no el botón de minimizar.
- **Movable**: El que la forma pueda o no ser movida del lugar en el que se encuentra.
- **Picture**: Tapiz que utilizará la forma.
- **Showtips**: Permitir o no visualizar los Tips de los objetos contenidos en ella.

- **Windowstate**: Estado en el que se encontrará la forma (*Normal*, Minimizada, Maximizada)

- **Windowtype**: El tipo de ventana de que se trata (*Sin modo*, Modal). En este caso el que se encuentre sin modo nos permitirá que el usuario pueda cambiarse de ventana o de aplicación sin haber terminado con lo que estaba haciendo. Utilizando el tipo Modal no permitirá que el usuario se cambie de ventana hasta que no halla cerrado correctamente en la que está.

Label

- **Alignment**: Alineación de la etiqueta (*Izquierda*, derecha, centro) con respecto al área en la que fue definida.

- **Autosize**: Que el área definida se ajuste o no al tamaño del texto.

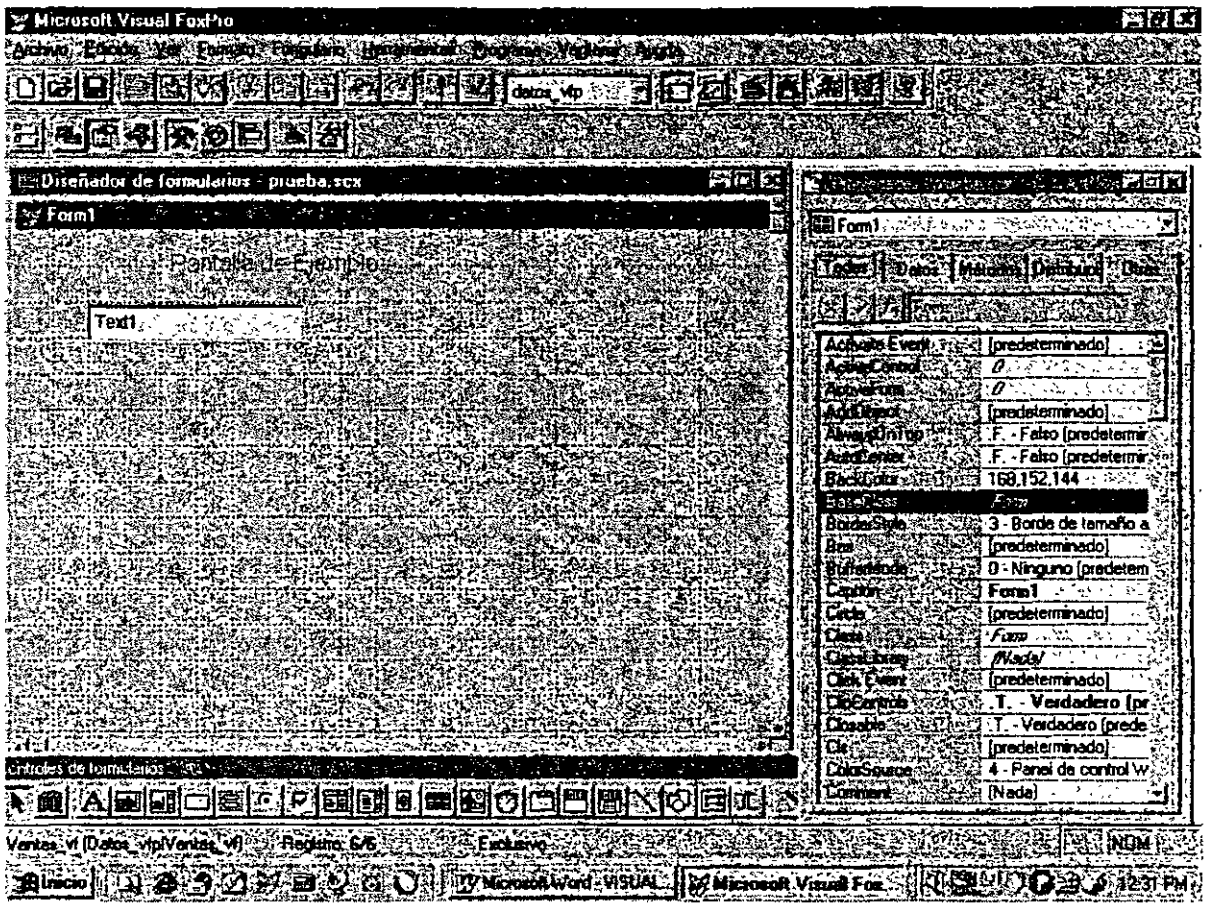
- **Backstyle**: El estilo del fondo (*Opaco*, Transparente). Cuando es opaco se verá el área de la etiqueta definida con otro color o con un marco, para el caso de transparente no se verá mas que el texto.

- **BorderStyle**: Se le define si se requiere marco en la etiqueta o si se omite.

Cuadro de texto

- **Century**: Activar o desactivar el manejo del siglo en campos tipo fecha.

- **DateFormat**: Formato que se utilizará para los campos tipo fecha.



- **Format**: Formato de entrada y salida de los datos.
  A continuación definimos los formatos que se utilizan:

  A Sólo permite caracteres alfabéticos (sin espacios ni signos de puntuación).
  D Utiliza el formato actual de SET DATE.
  E Edita los datos tipo Fecha como fecha BRITISH.
  K Selecciona todo el TextBox cuando se mueve el cursor al TextBox.
  L Muestra ceros a la izquierda (en lugar de espacios) en el TextBox. Utilice sólo datos numéricos.
  M Permite múltiples opciones preestablecidas. La lista de opciones se almacena en la propiedad InputMask como una lista de elementos delimitados por comas. Los elementos individuales de la lista no pueden incluir comas. Si la propiedad Value del TextBox no contiene uno de los elementos de la lista, se establecerá como el primer elemento de la lista. Se usa sólo con datos de caracteres. Sólo se aplica al TextBox.
  R Muestra una máscara de formato para el TextBox. Los caracteres de máscara no se almacenan en el origen del control. Se usa sólo con datos de caracteres o numéricos. Sólo se aplica al TextBox.
  T Recorta los espacios en blanco iniciales y finales del campo de entrada.

! Convierte a mayúsculas los caracteres alfabéticos. Se usa sólo con datos de caracteres. Sólo se aplica al TextBox.
^ Muestra los datos numéricos utilizando la notación científica. Sólo se usa con datos numéricos.
$ Muestra el símbolo de moneda. Sólo se usa con datos numéricos o de moneda.

- **Hours**: Tipo de reloj manejado de 12 o 24 horas.

- **Inputmask**: Muestra como es posible introducir la información dentro de un text.
  Es posible utilizar el siguiente código.

  X Puede introducirse cualquier carácter
  9 Pueden introducirse dígitos y signos, como el signo menos (-).
  # Pueden introducirse dígitos, espacios en blanco y signos.
  $ Muestra el símbolo de moneda actual (especificada con SET CURRENCY) en una posición fija.
  $$ Muestra un símbolo de moneda flotante que siempre aparece junto a los dígitos del Spinner o TextBox.
  * Se muestran asteriscos a la izquierda del valor.
  . Un punto especifica la posición de coma decimal.
  , Pueden incluirse comas para separar dígitos a la izquierda de la coma decimal.

- **PasswordChar**: Carácter que se utilizará para campos en los que no se quiere mostrar el contenido de lo capturado, por ejemplo: Password.

- **Readonly**: Que el dato mostrado sea de solo lectura.

- **Value**: El valor que posee inicialmente el campo o con el que lo inicializamos.

Cuadro de edición

- **MaxLength**: Máxima longitud del texto a escribir en la región de edición

- **ScrollBars**: Las barras de desplazamiento que mostrará en la ventana de edición. (Ninguna, *Vertical*)

Botón de comando

- **Picture**: Será la imagen que le pegaremos al botón.

- **WordWrap:**

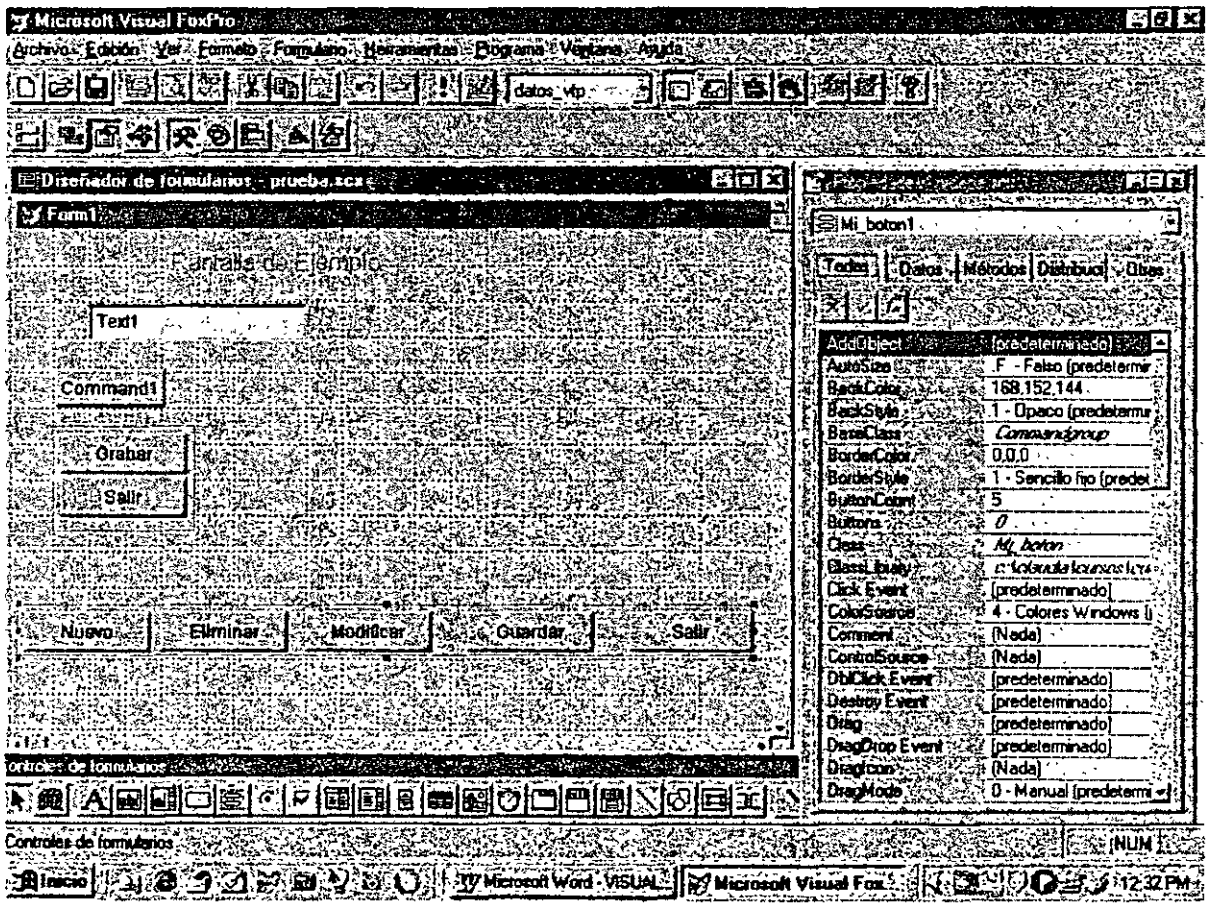
Grupo de comandos

Como grupo tenemos:

- **ButtonCount**: Número de botones que tendrá el grupo.

Grupo de opciones

Igual al grupo de comandos

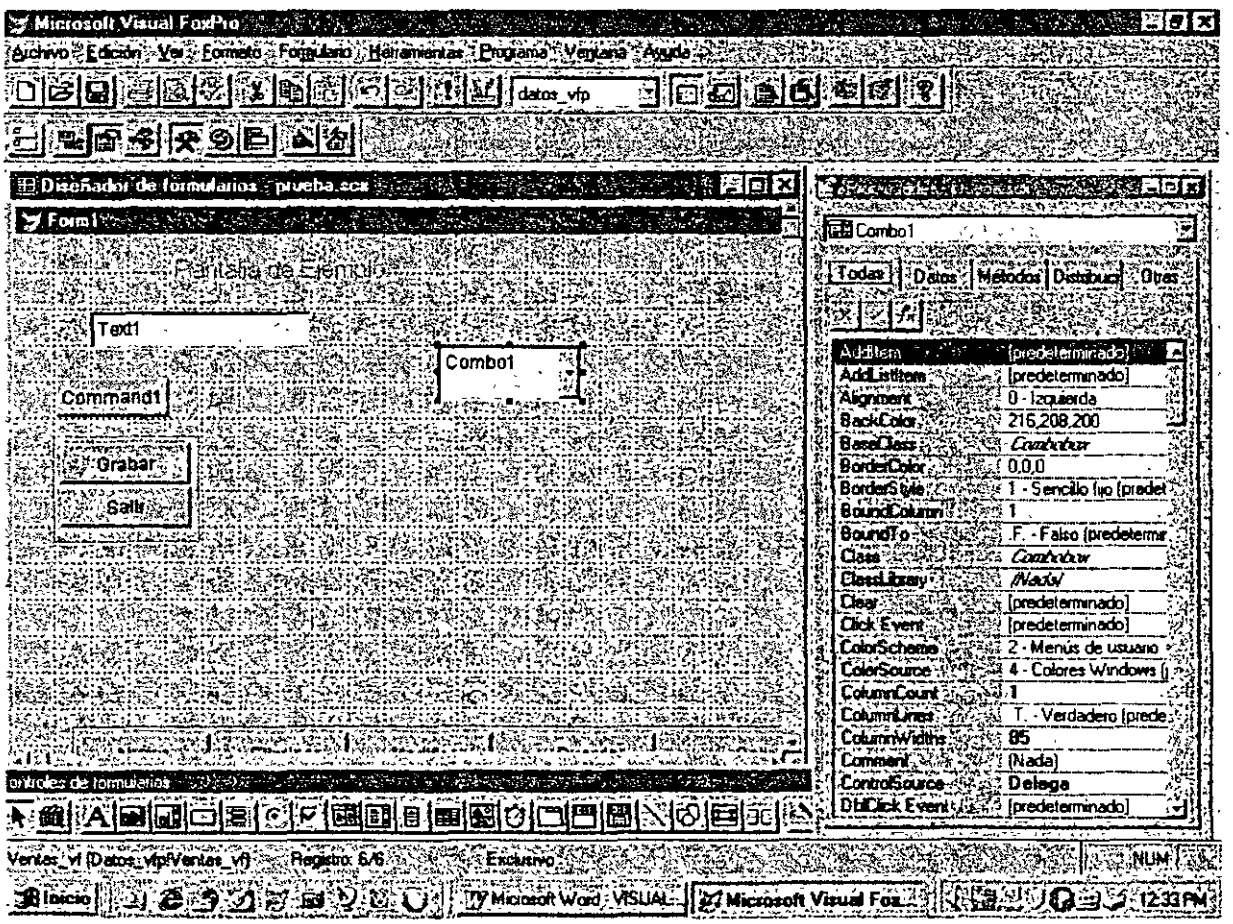


Cuadro combinado

- **ColumnLines**: Si muestra o no las líneas de división entre las columnas de la lista.
- **ColumnWidths**: Ancho de las columnas.
- **FirstElement**: Primer elemento que se mostrará de la lista contenida en el cuadro.

- **ItemBackColor**: Color del fondo de la lista
- **ItemForeColor**: Color de la letra utilizada en la lista.
- **ItemTips:** Muestra ayuda o no de las opciones de la lista.
- **Rowsource**: El contenido de los renglones de la lista. Pueden ser los campos de un campo, datos ingresado a mano o datos que se obtienen de un arreglo definido por el usuario.
- **RowsourceType**: Tipo de dato almacenado (Valor, alias, inst. SQL, consulta, matriz, campos, archivos)
- **Style:** Tipo de despliegue como lista o cuadro.

Para el caso de este tipo de lista se verá que es más fácil manipularlos mediante el constructor. El constructor nos mostrará la siguiente pantalla. Esta nos irá guiando paso a



paso.

Cuadro de lista

- **MoverBars**: Aparezcan un marcado de desplazamiento para los elementos de la lista, permitiendo al usuario reordenarlos.

Control numérico

- **Increment**: Incrementos de cuanto irá aumentando el contador numérico.
- **KeyboardHighValue**: El valor mas alto que se puede introducir por teclado.
- **KeyboardLowValue**: El valor mas pequeño que se puede introducir por teclado.
- **SpinnerHighValue**: El valor mas alto que se puede obtener mediante las flechas de desplazamiento.
- **SpinnerLowValue**: El valor mas bajo que se puede obtener mediante las flechas de desplazamiento.

Cuadrícula

- **AllowAddNew**: Especifica si es posible anexar registros desde una cuadrícula.
- **AllowHeaderSizing**: Permite modificar el tamaño de los encabezados.
- **AllowRowSizing**: Permite modificar el tamaños de los renglos.
- **Columncount**: Número de columnas que va a contener la cuadrícula.
- **Deletemark**: Marca de borrado. Cuando desde un grid es posible marcar para borrar un registro. Esto mediante la columna de borrado de la tabla.
- **Gridlinecolor**: El color de las líneas de división de la cuadrícula.
- **Gridlinewidth**: Ancho de las líneas de división de la cuadrícula.
- **Gridlines**: Las líneas de división de la cuadrícula. (Ninguna, Horizontal, Vertical, *Ambas*)
- **Headerheight**: Alto de la barra de encabezados.
- **Highlight**: El que se identifique el registro que se encuentra seleccionado mediante otro color.

- **Recordmark**: Que aparezca o no la marca de registro.
- **Rowheight**: Alto de los renglones.
- **SplitBar**: En caso de que se desee que aparezca o no la barra de división
- **View**: Como mostrará la información, si se permite lectura y escritura o solo escritura (*Examinar-Examinar*, Examinar-Cambiar, Cambiar-Examinar, Cambiar-Cambiar)

Recordemos que la cuadrícula está compuesta por cuatro partes: la cuadrícula, columnas, encabezado y campo.

Las propiedades anteriores sólo engloban a la cuadrícula. Veremos las de las columnas

Columnas

- **Bound**: Indica si existe una relación entre la variable de la columna y el dato almacenado. Cuando el Bound es verdadero no es posible manejar por separado los datos de la columna y del objeto texto. Esto resulta lo mas conveniente.
- **ControlSource**: La variable donde se almacenará el valor.
- **Resizable**: Permitirá que el usuario pueda modificar el tamaño de las columnas.

Encabezado

Sólo se manejará como si fuera un texto o etiqueta.

Texto

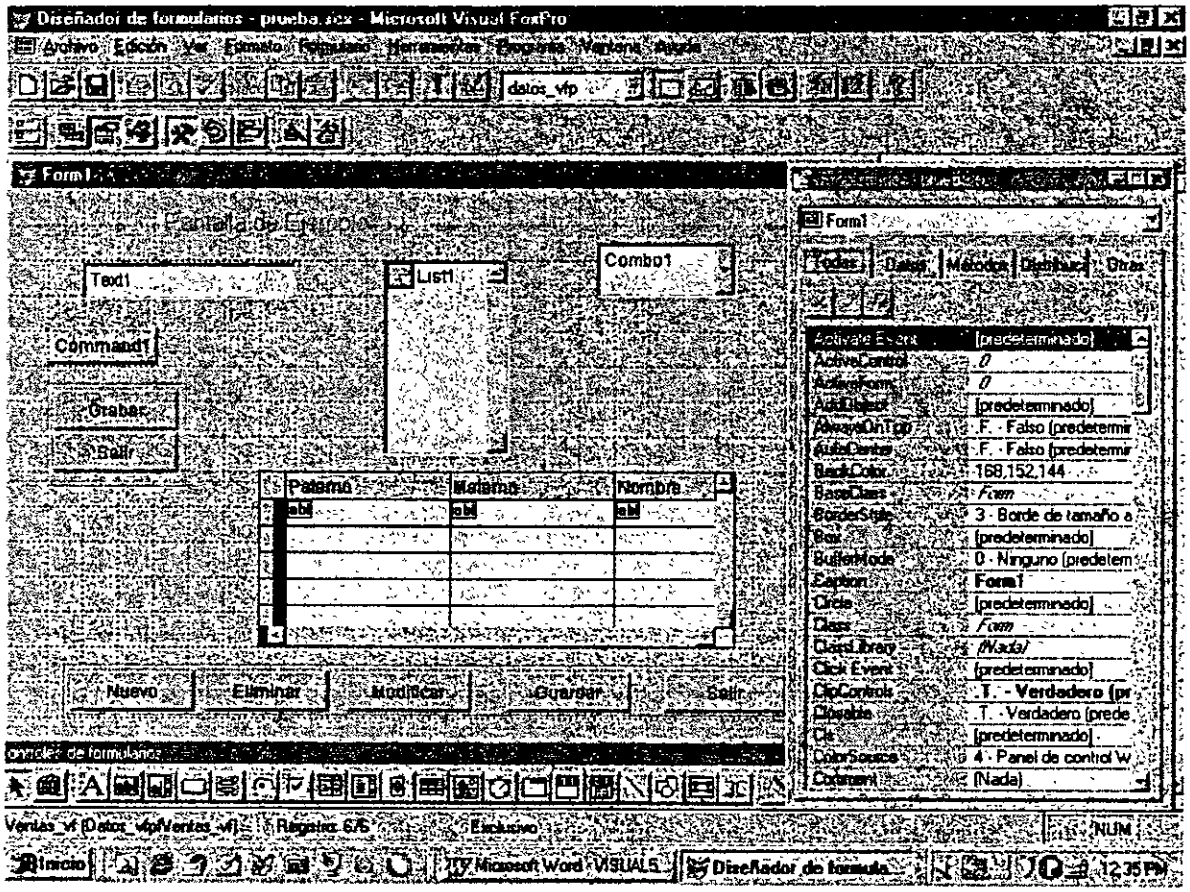
Es exactamente igual que el objeto cuadro texto.

Marco de página

- **PageCount**: Número de páginas que contendrá el marco.
- **ActivePage**: Página que estará activa al abrir el marco.

Al igual que en otros objetos el marco de página es un grupo de páginas, por lo que de igual manera se trabajará como conjunto y de forma individual.

Nota: Todos aquellos objetos que estén contenidos por otros será necesario marcarlos, oprimir botón derecho del mouse y seleccionar EDITAR para poder tener acceso a los elementos individuales o por medio de la ventana de las propiedades



CAPITULO V.
Manipulación y programación de Clases y eventos

En este capítulo trabajaremos con lo poco de código que se utilizará dentro de la programación orientada a objetos.
La dificultad que se podría encontrar sería quizá la distribución del código dentro de los diferentes métodos.
El programador ahora deberá pensar de acuerdo a acciones que le pueden suceder al objeto.
Veremos primero la definición de cada método, al igual que en las propiedades existen métodos genéricos y específicos y de la misma manera los iremos tratando.

¿Cómo hacer referencia a un objeto?

Para llamar a un objeto dentro de cualquier método se seguirá la siguiente estructura:

Dominio.Objeto padre.Objeto.Propiedad=Valor

Dominio.Objeto padre.Objeto.Método

Donde :

Dominio: Es la forma sobre la cuál se encuentra el objeto, la llamaremos por su Name o en forma general THISFORM

Objeto Padre: El objeto del que depende (en el caso de objetos compuestos) por ejemplo: Un botón de comandos dentro de un grupo de comandos. En este caso se determinaría el objeto padre como el CommandGroup y al objeto como el Command, quedando:

Thisform.CommandGroup1.Command3.Setfocus

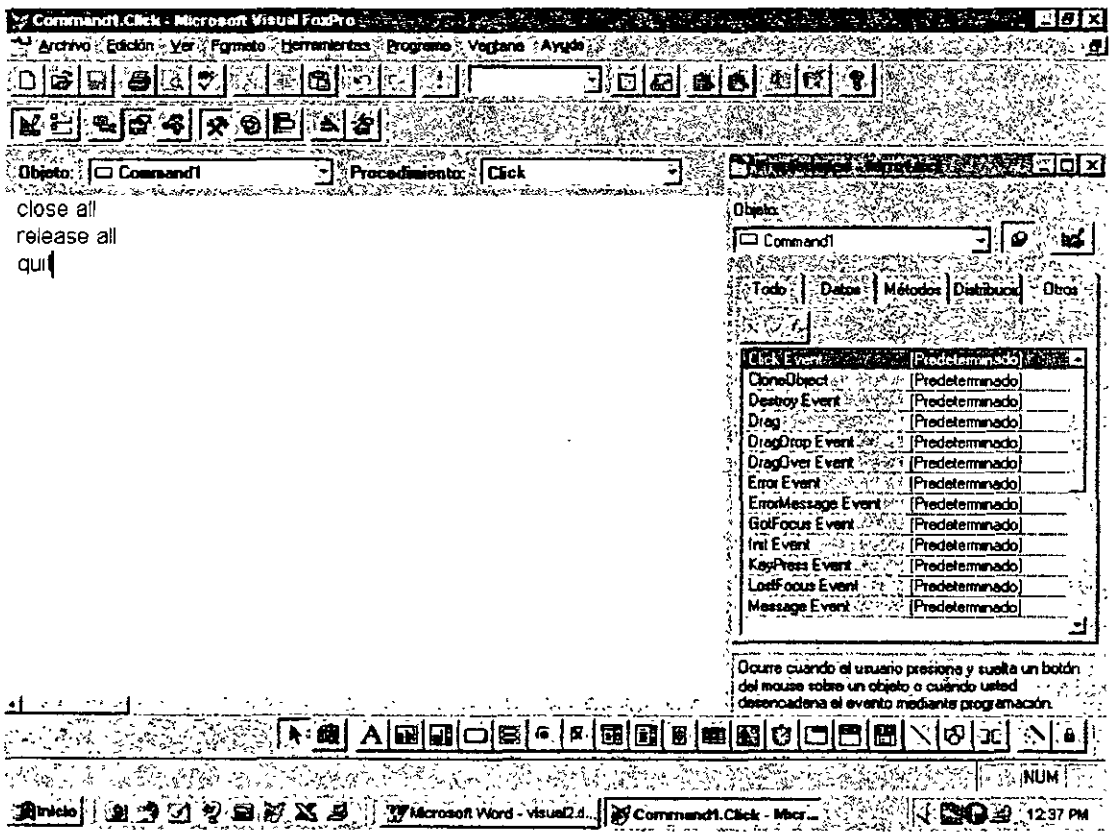
Para el caso de objetos mas complejos como el marco de página tendríamos:

Thisform.PageFrame1.Page1.Command3.Setfocus

Objeto: El objeto que recibirá finalmente la instrucción.

Métodos Generales

- **Click**: Cuando sensibiliza el sistema un click del mouse sobre el objeto. Pensemos que se tiene un Command Button o botón de comandos, él cuál al oprimirse cerrará tablas, desactivará y eliminará de memoria las ventanas activas. Estas acciones deseamos sucedan en el momento de darle click con el mouse.
  Primer paso: Llamaremos al evento Click del botón de comandos, esto mediante un doble click sobre el método dentro de la ventana de propiedades. Al hacerlo se nos mostrará la pantalla siguiente:



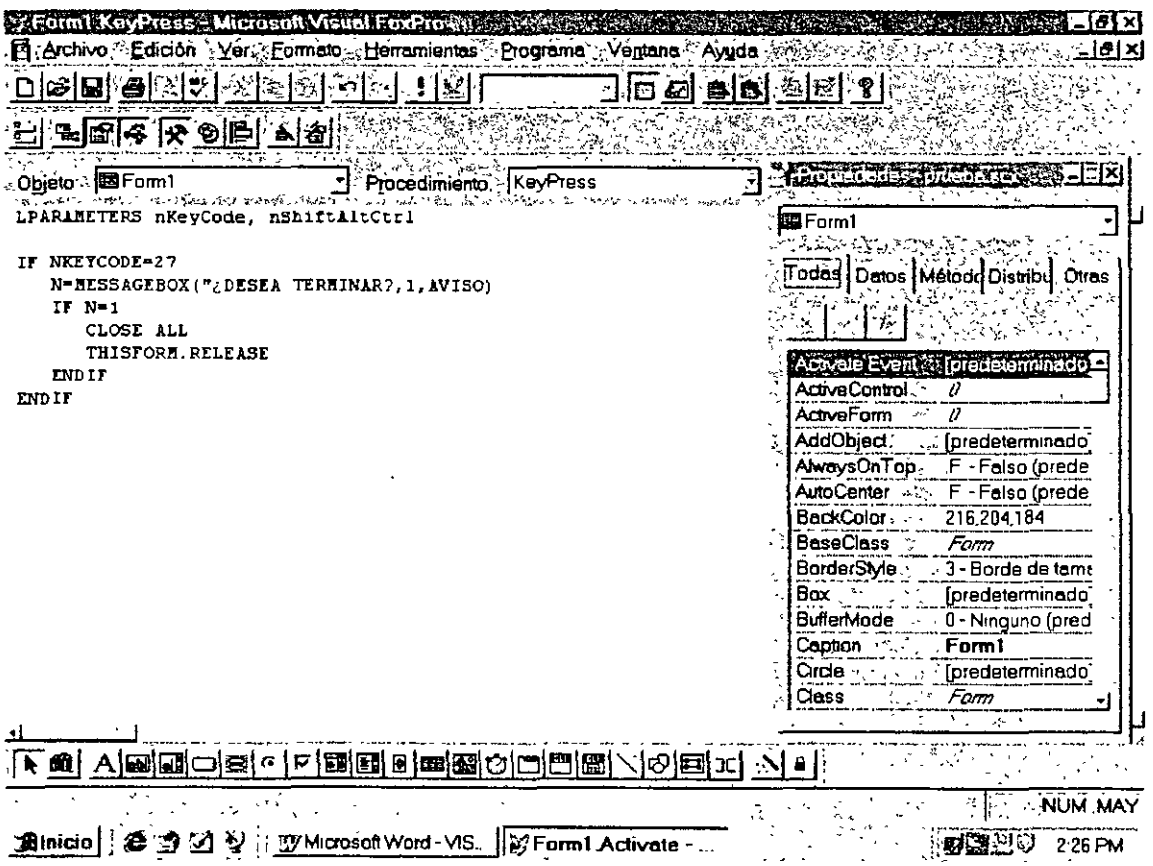
Analizando la pantalla veremos que en la parte superior izquierda nos indicará el nombre del objeto y del lado derecho el procedimiento. Aquí es donde debemos de incluir el texto correspondiente a los procesos que deseamos realice.

- **Dobleclick**: En este caso es cuando sensibiliza un doble click el objeto.

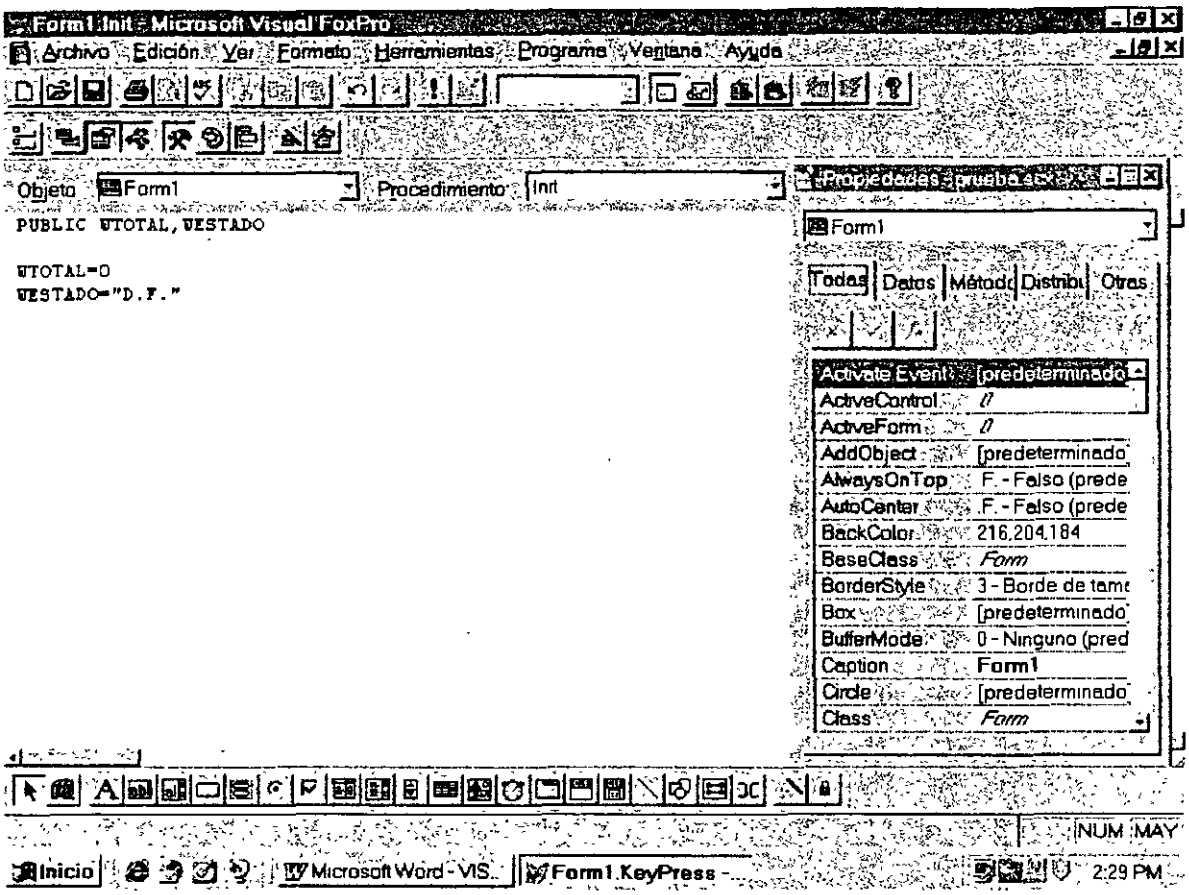
- **RightClick**: En este caso es cuando sensibiliza que el botón derecho del mouse es el que está realizando la llamada al método.

- **MiddleClick:** Llama al método al sensibilizar el botón medio del mouse.

- **KeyPress:** Cuando el objeto detecta que se ha oprimido una tecla siempre recorre al evento Keypress para verificar si no es una tecla programada. Veamos que sucedería si en el caso de oprimir ESC. Quisiéramos que enviara un mensaje a pantalla preguntando si el usuario desea salir del proceso que se está realizando.
  En algunos casos los procesos nos proporcionarán parámetros que el Visual los hace viajar a éste para comodidad en la manipulación de la información.
  Para el evento Keypress tenemos dos parámetros: nKeyCode, nShiftAltCtrl. El primero nos proporcionará el código de la tecla que se oprimió (ASCII) y el segundo el valor para el caso de que utilicemos teclas combinadas con las de control.
  Tendríamos:



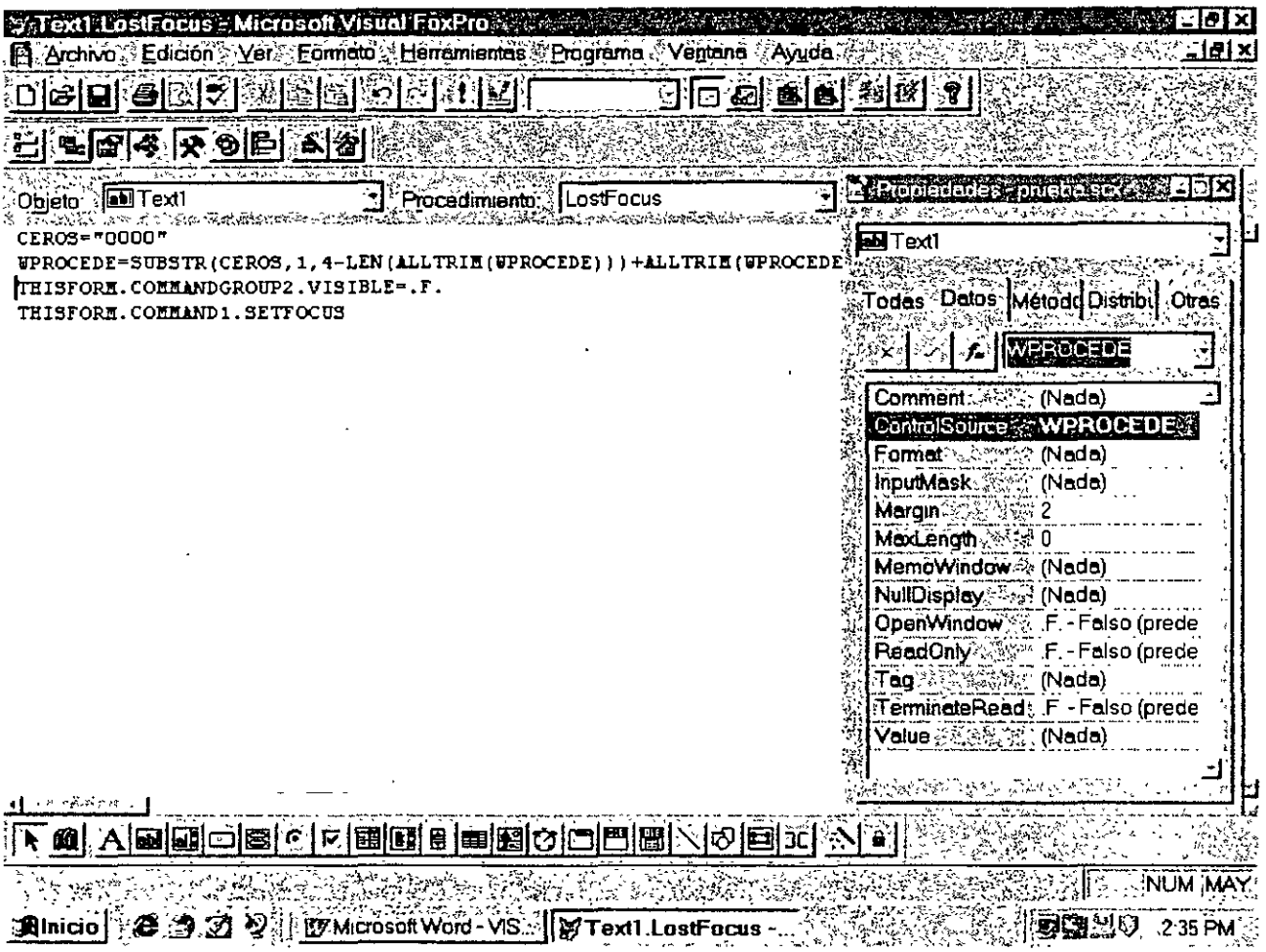
- **Init**: Son todos aquellos procesos, definiciones e inicializaciones de variables que se requiere se procesen en el momento que se abre la forma. Es aquí donde declararemos variables públicas, para el caso de las formas.



Los siguientes eventos son de los mas importantes en cuanto a su manejo:

- **GotFocus**: Es el momento en que el objeto se encuentra en foco esto es que, la atención de Visual se encuentra puesta en él. Este método es útil para realizar funciones que requerimos cuando nos encontramos dentro de un objeto.

- **LostFocus**: A diferencia del anterior el Lostfocus es cuando el objeto acaba de perder el foco esto es, ahora la atención de Visual se encuentra puesta en otro objeto. Este método lo utilizaremos para programar acciones que dependerán del valor del objeto que acabamos de abandonar. Por ejemplo, en la forma de captura existe un área donde le determinamos si el cliente es local o foráneo. Para el caso de foráneo necesitamos se llene un dato correspondiente al estado y país de origen, pero si es local no. Entonces cuando el objeto donde le determinamos el origen del cliente pierde el foco le se validará la condición anterior, teniendo lo siguiente:

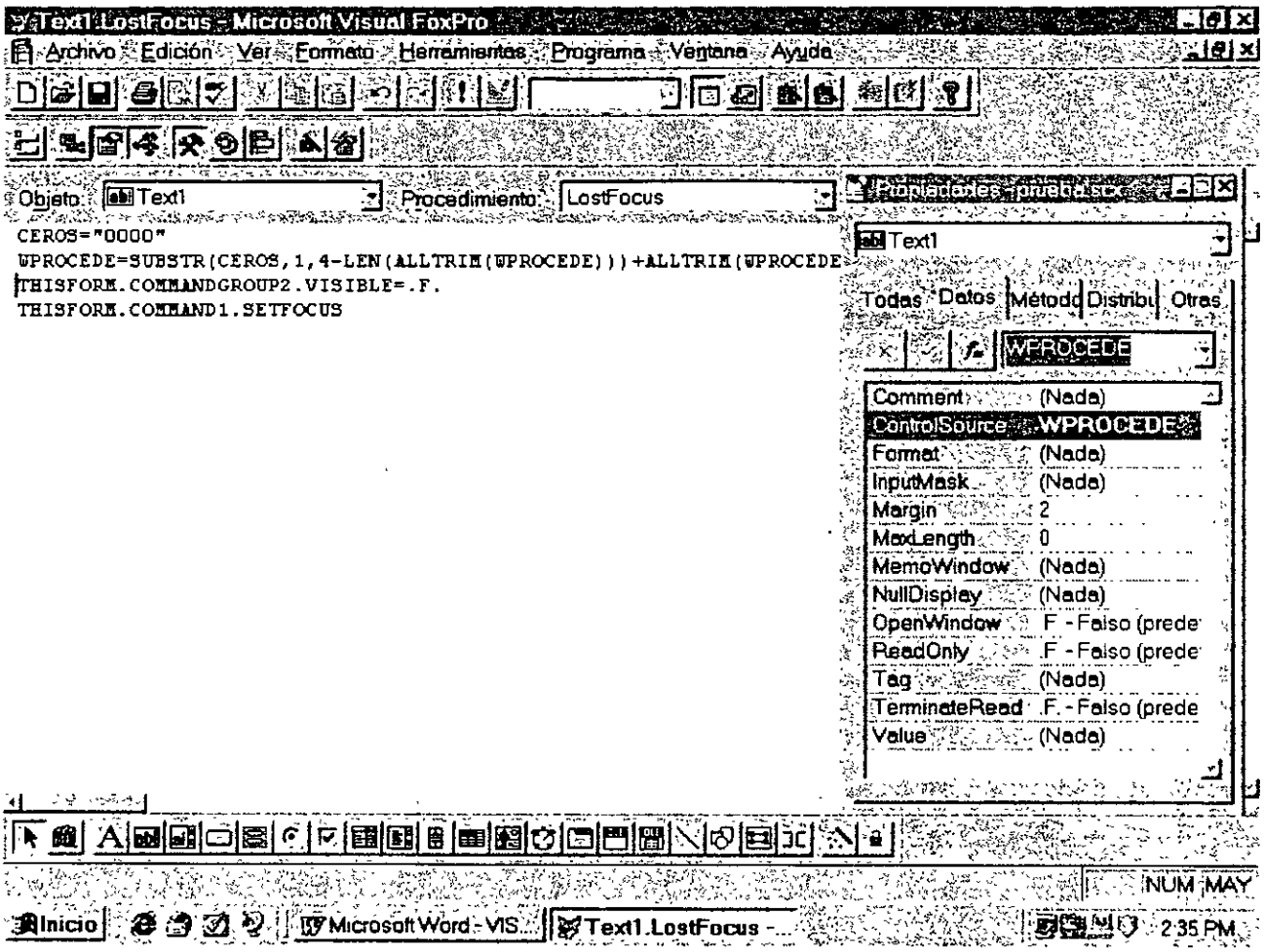


Aquí wprocede será la variable que almacena el valor del objeto capturado, en el evento de LostFocus es el lugar donde se le dará formato, ya que es aquí cuando el usuario tecleó su opción y se oprimió enter para continuar.
Como se observa después de formatear modificará la propiedad de Visible del objeto Commandgroup y enviar el foco al Command1. Estas propiedades serán modificables mediante código desde un evento. Recordando que al hacer la llamada al objeto debemos de indicarle a quienes pertenece. En este caso pertenece a un grupo y este a una forma.

Métodos Específicos

Forma

- **Activate**: Es aquel método que se ejecuta cuando la forma se abre, se ejecutará cuantas veces se vuelva a activar a la forma, a diferencia del Init que sólo se realiza una vez (la primera vez que se abre)



Aquí wprocede será la variable que almacena el valor del objeto capturado, en el evento de LostFocus es el lugar donde se le dará formato, ya que es aquí cuando el usuario tecleó su opción y se oprimió enter para continuar.
Como se observa después de formatear modificará la propiedad de Visible del objeto Commandgroup y enviar el foco al Command1. Estas propiedades serán modificables mediante código desde un evento. Recordando que al hacer la llamada al objeto debemos de indicarle a quienes pertenece. En este caso pertenece a un grupo y este a una forma.

Métodos Específicos

Forma

- **Activate**: Es aquel método que se ejecuta cuando la forma se abre, se ejecutará cuantas veces se vuelva a activar a la forma, a diferencia del Init que sólo se realiza una vez (la primera vez que se abre)

- **Release**: Cuando deseamos abandonar la forma y eliminarla de memoria.
- **Refresh**: Refresca a la forma, esto es renueva los valores de las variables.

Cuadro de texto

- **ProgrammaticChange**: Este evento se programa cuando el valor del objeto está en constante cambio, por lo que el evento se dispara al sensibilizarse esto.
- **SetFocus**: Es cuando cualquier objeto envía el foco a otro. Ahora veamos a los tres eventos relacionados con el foco.

FOCO

SE TIENE (GOT) PIERDE (LOST) ADQUIERE (SET)

Estos son tres momentos importantes de un objeto ya que dependiendo es lo que podemos hacer o manipular.

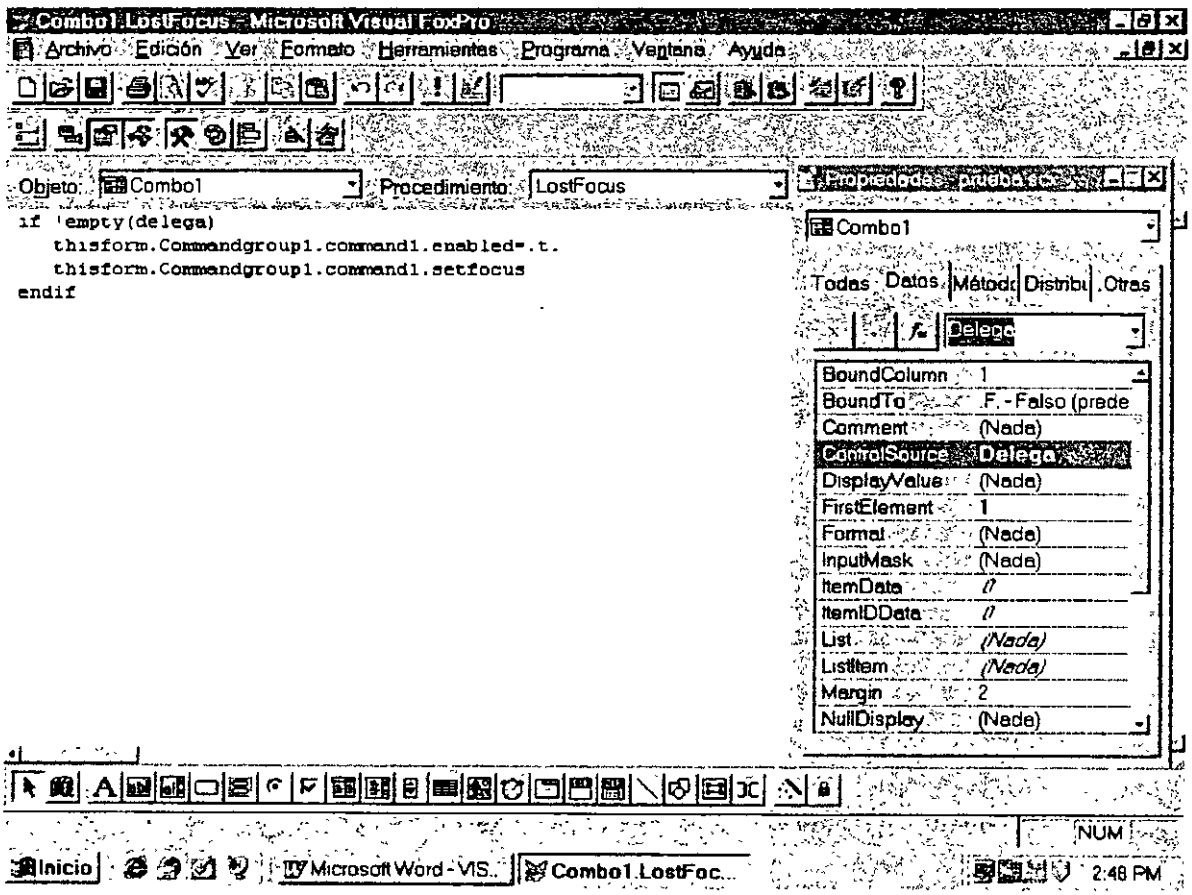
- **Valid**: Es el método que nos permitirá realizar la validación de una entrada, esta puede ser tan compleja como lo requiera el desarrollo
- **When**: Este método es una pregunta donde podremos condicionar salidas, búsquedas despliegues de información, etc. Podríamos decir que traduciéndolo sería Cuando suceda esto haz esto otro.

V.1 Propiedades y eventos

Dentro de los métodos la programación será muy semejante a la conocida en cualquier lenguaje Xbase, pero también es el lugar donde modificaremos propiedades y enviaremos a un objeto a algún evento en especial.

Trabajemos con ejemplos:

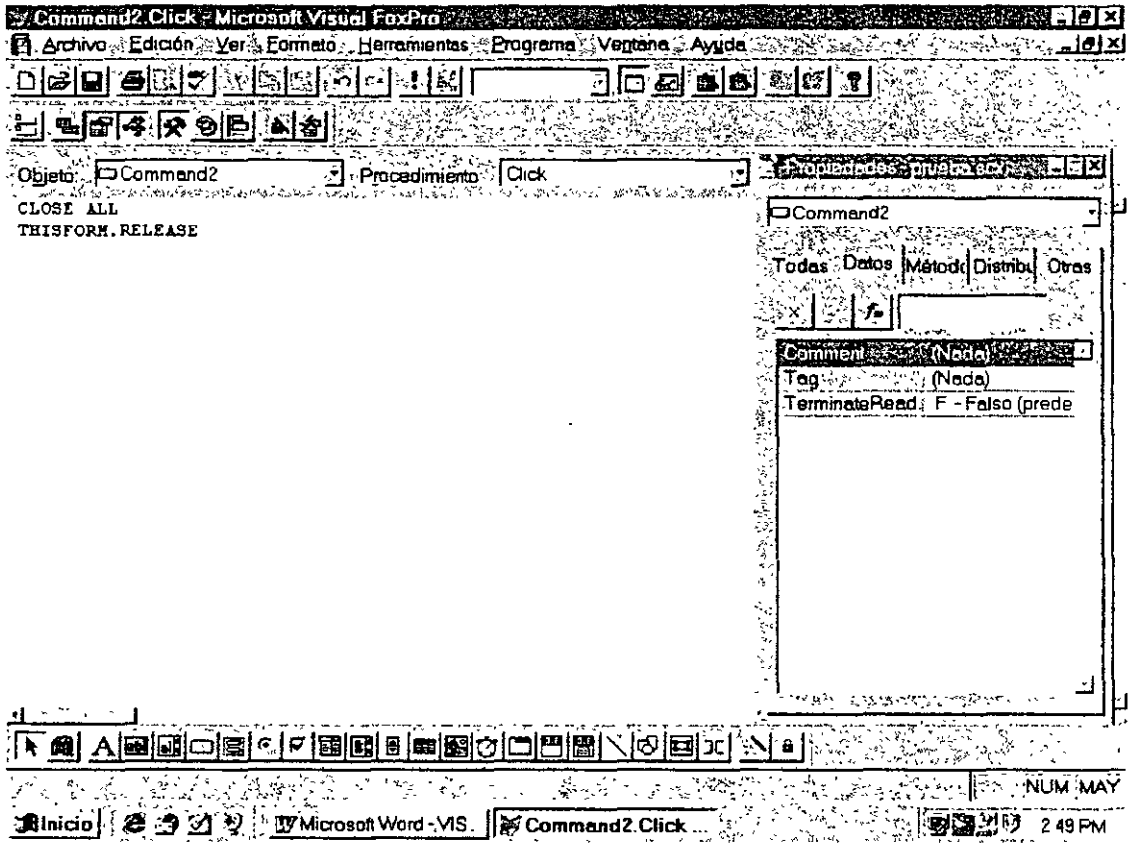
Pensemos que tenemos una barra de herramientas propia del sistema, donde programamos a cada botón dependiendo de las necesidades del proyecto. Dentro de estos botones poseemos uno de guardar y otro de salir.
El botón de guardar sólo deseamos que se active cuando el capturista ha terminado de teclear la información mas relevante del registro, por lo que al abrirse la forma la propiedad de habilitado del botón se encuentra en Falso. El botón no se podrá oprimir aún cuando sea visto. Cuando el programador considera que llenó la información suficiente se habilita para poder grabar. Veremos:



Por lo que podemos observar en las propiedades, Enabled del command se encuentra en Falso, propiedad que se modificará en el método Lostfocus del cuadro de texto que requiero sea llenado.
Notemos que nuevamente en el código tenemos que hacer referencia al dominio del objeto, en este caso lo último a escribir será la propiedad del objeto que se desea modificar y el valor que deberá de adquirir.

Para el caso de los métodos existen algunos que pueden ser referidos y actuarán en forma directa. Por ejemplo, tenemos que el evento Click del botón de salida que hará una llamada al evento RELEASE. Esto generará que la forma se cierre y desaparezca de memoria.

Veamos que los métodos de acceso directo sólo serán comandos a realizarse no poseen ningún valor.



En resúmen la diferencia entre las propiedades y los eventos que se pueden ejecutar o modificar desde código, es que en los primeros se igualará un valor a la propiedad correspondiente, la segunda solo se le hará referencia.

CAPÍTULO VI. Las Clases y Librerías

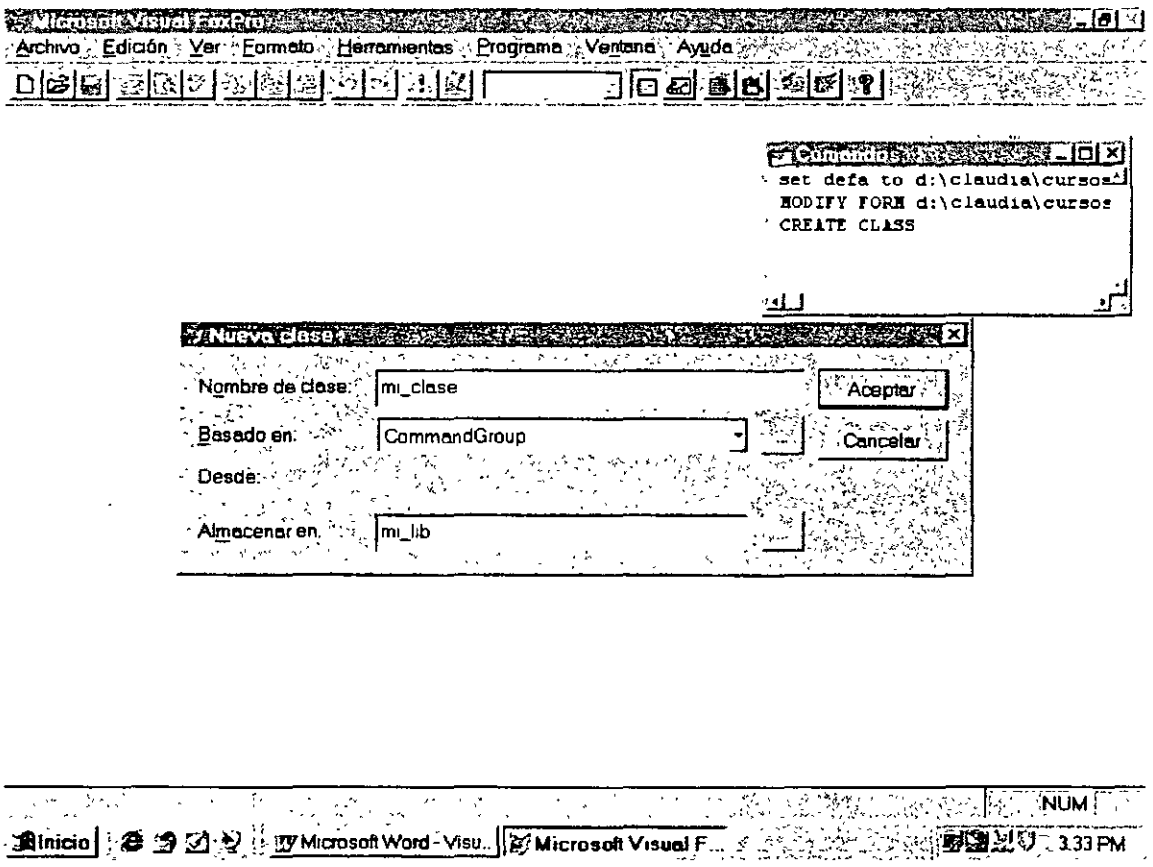
VI.1 Clases en Visual FoxPro

En Visual FoxPro se diseñará una clase muy parecido a como se trabaja con las formas. La diferencia radica en que cuando se guarda como clase, es posible seleccionarla de la barra de herramientas.

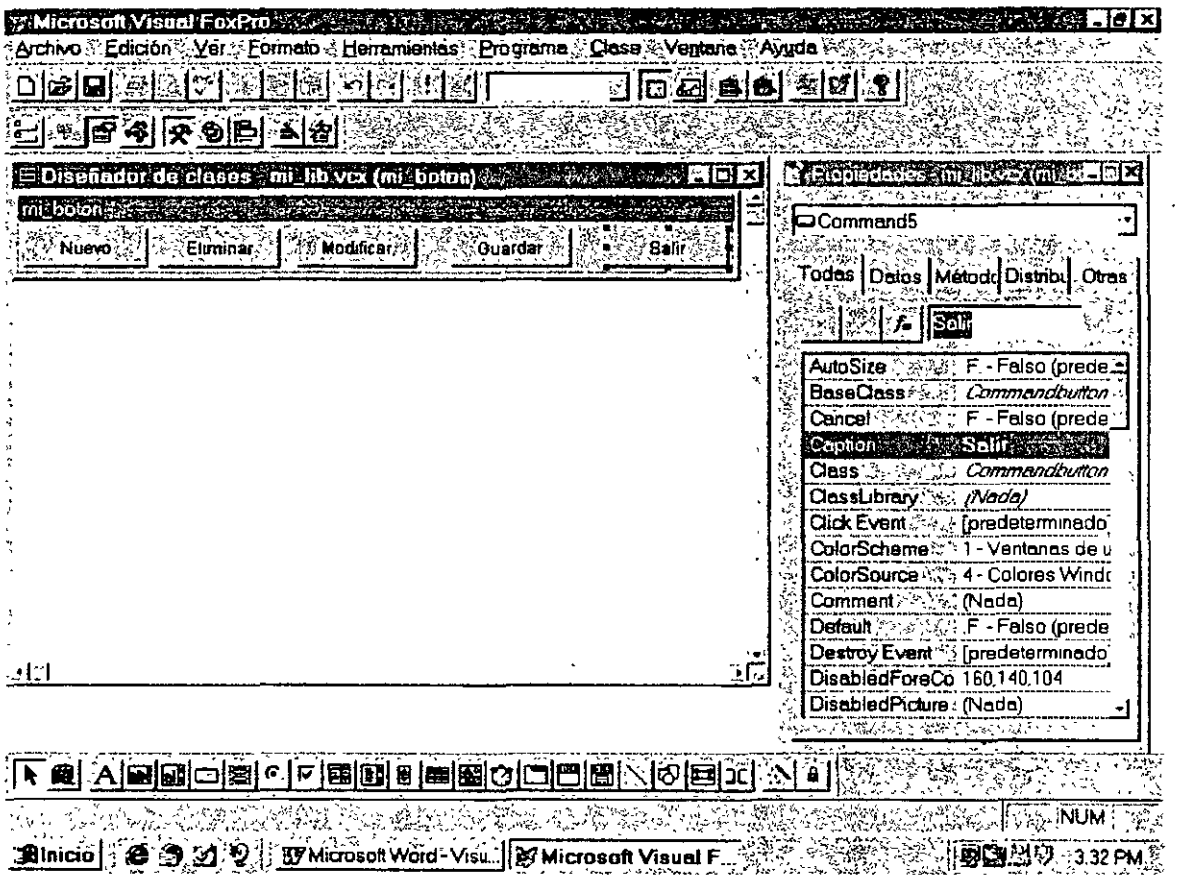
Cuando se crean subclases, éstas heredan todo el comportamiento de sus padres, éstos es todo el código de los eventos programados y propiedades.
La clase creada será almacenada en una biblioteca de clases.

VI.2 Creación de una clase

Para crear una clase debemos por medio de menú llamar a Archivo-Nuevo-Clase y el sistema nos mostrará la siguiente pantalla:



Donde nos preguntará el nombre de la clase, en que clase existente está basada y en la librería que lo almacenará.



Una vez que se le ha definido nombre, librería, etc. Se nos mostrará una pantalla parecida a la de la creación de una forma donde nos permitirá ir incluyendo los objetos. De acuerdo al objeto seleccionado es el tipo de objeto que podremos incluir. Por ejemplo, se realiza una barra de botones de procesos para una aplicación específica, esta quedaría de la siguiente forma:

Veremos que la barra es el único espacio donde aparecen objetos.

¿Qué sucedería si se desea crear una clase basado en la clase Forma? Estaríamos creando una plantilla de formas que utilizaremos en mas de una ocasión en donde agregaremos aquellos objetos que irán siempre fijos.

A la clase se le definirán propiedades específicas y podrán programarse acciones determinadas. Por ejemplo, que al oprimir el botón de salir cierre todo y salga, como crear un nuevo registro, etc.

Una vez definida la clase, sus propiedades y métodos lo que procederá es anexarla a las formas. Este proceso nos ahorrará mucho trabajo, ya que el código será tecleado sólo una vez y será anexado muchas.

VI.3 Librerías o bibliotecas

Llamaremos librería al conjunto de clases. Pensemos que es un cajón donde iremos almacenando todos aquellas herramientas nuevas que estamos creando. Dentro de una librería es posible almacenar una serie de clases creadas.
Las clases serán referenciadas dentro de la clase por medio del nombre con el que fueron creadas.
Es recomendable que dentro de las librerías se almacenen clases relacionadas entre sí. Por ejemplo, para el caso de que se creara una clase barra de herramientas, una menú, podríamos anexarla a una librería a la cual llamaremos Barras.

Ahora también es recomendable crearlas de acuerdo a la aplicación.

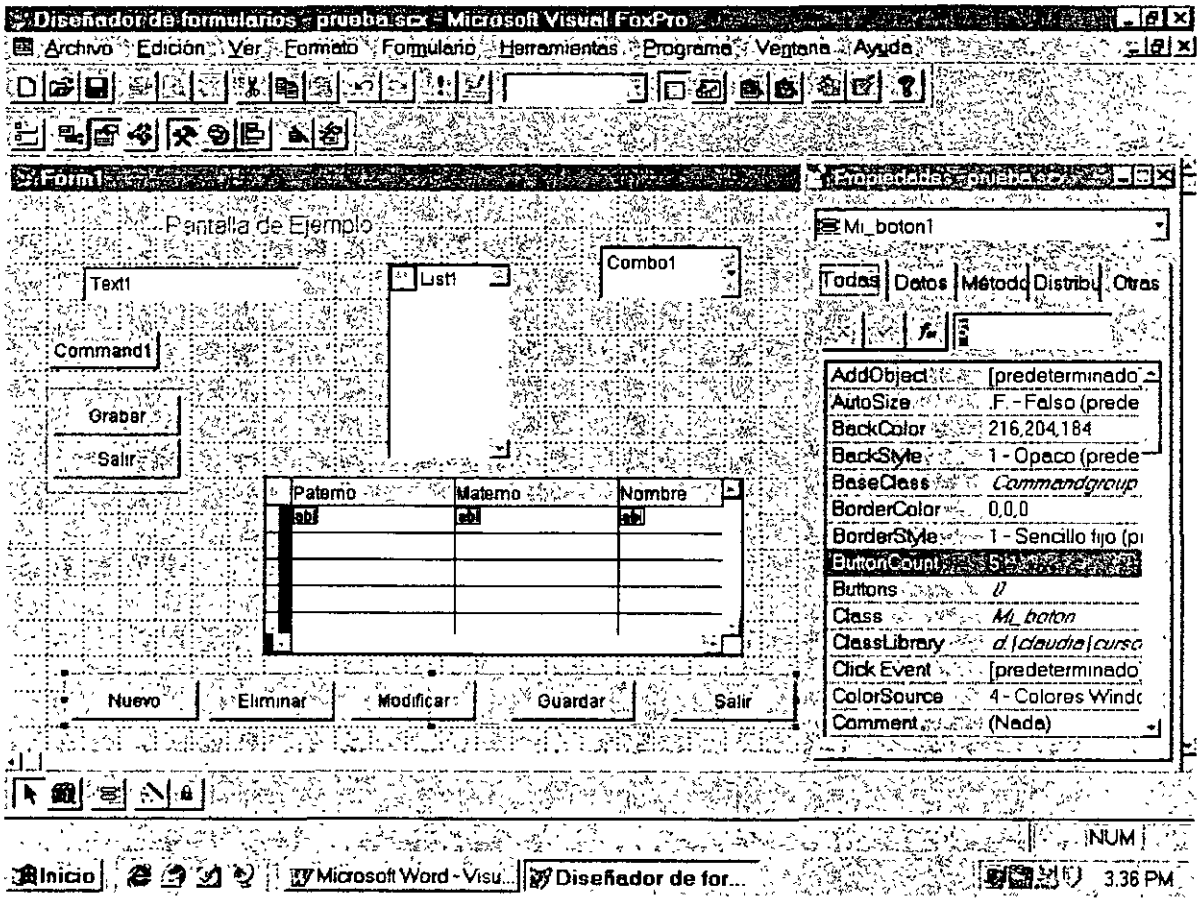
VI.3.1 Inclusión de una clase

Para anexar a una forma una clase creada es necesario hacer lo siguiente:

Dentro de la barra de herramientas se encuentra un botón donde muestra unos libros (ver clase). Al oprimir el botón aparecerá un menú: Agregar, *Estándar*, Controles OLE.
La librería estándar es la que se muestra originalmente por lo debemos de seleccionar Agregar.

Al elegir agregar se nos mostrará la ventana para seleccionar un archivo, aquí veremos las librerías (archivos VCX) existentes.
Una vez hecho esto la barra de herramientas se verá modificada de acuerdo a las clases que posee la librería activa. Por lo que los botones cambiarán.

Veremos la siguiente pantalla, en la cuál se muestra la barra de herramientas modificada y la nueva clase anexada a ella:



Para poder obtener nuevamente la barra de herramientas estándar oprimiremos el botón de librerías pero seleccionaremos Estándar y nos mostrará la barra conocida, dejando en memoria la anexada.

CAPITULO VII. Reporteador

VII.1 Definición

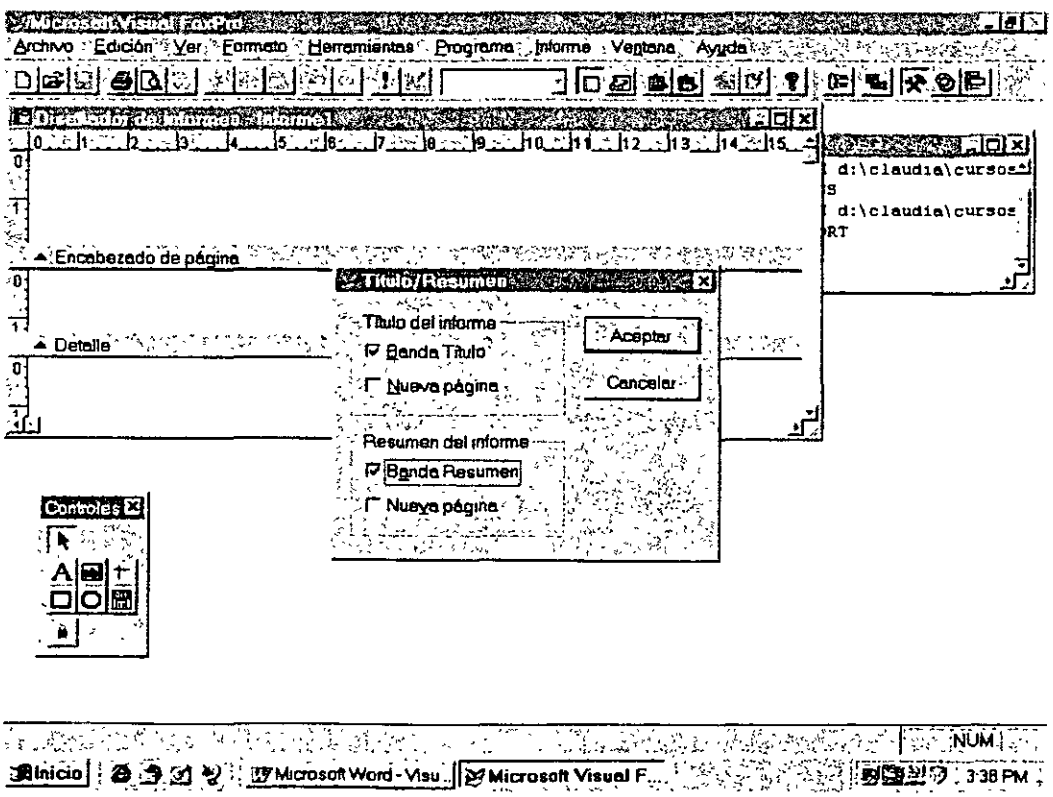
El generador de reportes es una herramienta que utilizaremos para crear reportes de nuestras bases de datos con gran diversidad, en la cuál será posible manipular la información agrupándola, obteniendo sumatorias, acumulados, etc.

VII.2 Generador de reportes

Llamaremos al generador por medio del menú Archivo-Nuevo-Informe y se añadirá una nueva opción en la barra del menú superior Informe, presentándose la siguiente pantalla:

Dentro del menú desplegable que presenta la opción de Informe, tendremos las siguientes opciones que definiremos a continuación:

Título/Sumario: Permite añadir bandas de títulos y sumarios en el reporte, al seleccionar esta opción aparecerá la siguiente pantalla:



Donde preguntará si se tendrán títulos y un resumen. Son páginas que se utilizan para obtener una hoja en donde se puntualizan datos finales y se tiene una hoja de presentación.

Agrupar datos: Cuando se desea crear grupos de información para obtener sumas, estadísticas, etc. Esto se verá con mayor detenimiento mas adelante.

Variable: La creación de variables de trabajo dentro del informe. Por ejemplo, el resultado de una suma.

Fuente predeterminada: Si se define un Font predeterminado para todo el informe.

Sesión privada: Se maneja para el trabajo en sistemas en red, el que el informe sólo lo ejecute el primero que hizo la llamada.

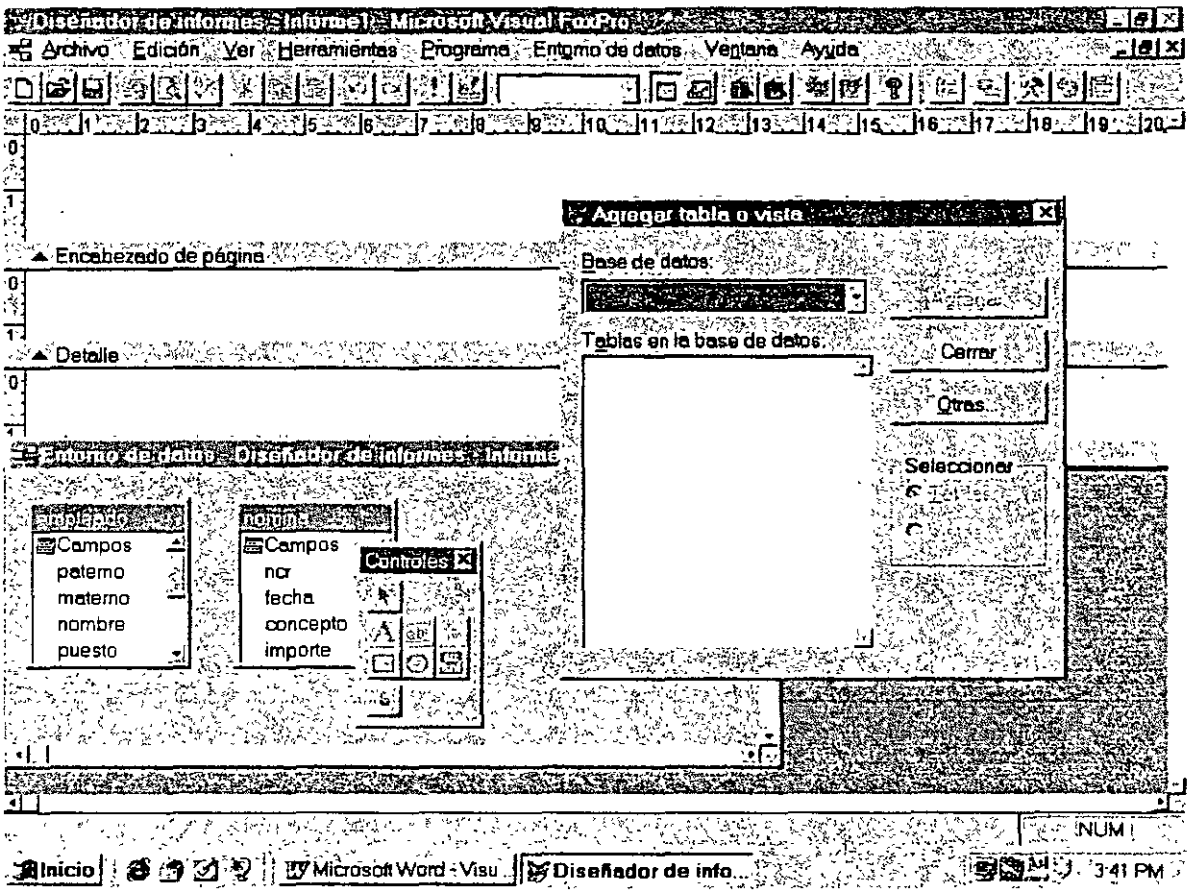
Informe rápido: Crea un reporte con todos los datos que posee la tabla incluída, sin que el programador intervenga.

Ejecutar: Ejecuta el informe.

Es de mencionar que ahora dentro del reporte no aparecerán opciones de impresión ya que todo se manejará a través del menú de Archivo en las opciones Preparar página e impresión preliminar.

VII.3 Entorno de datos

Dentro de los informes debemos de incluir dentro del entorno de datos, todas aquellas tablas que se utilizarán dentro de este.
Mediante el botón derecho del mouse haremos la llamada al entorno de datos y ahí agregaremos las tablas correspondientes.



Llamando a las propiedades de cada tabla modificaremos aquellas que sean útiles en nuestro desarrollo como podrían ser Order, para indicarle que estará indexada y por cuál orden lo estará.

VII.4 Barra de herramientas

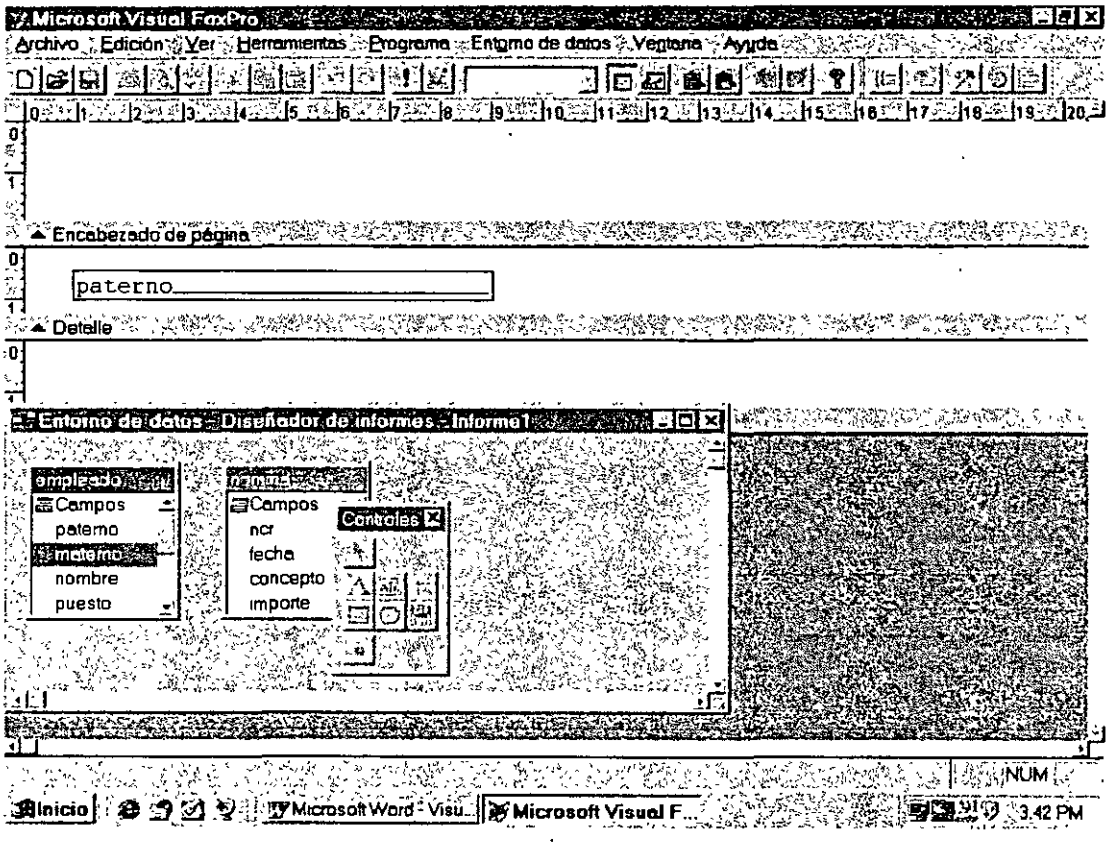
Encontraremos en la pantalla una barra herramientas que proporciona el constructor para ayudar a un rápido acceso a ciertos procesos.
Para insertar un objeto a un reporte ha de hacer click primero sobre la herramienta solicitada y después arrastrar hasta la ventana del reporte.

VII.4.1 Inserción de campos

Para insertar un campo dentro del reporte lo haremos mediante el botón marcado con dos letras minúsculas ab y después en la ventana del reporte marcar en donde se desee colocar o teniendo el entorno de datos abierto hacer Drag&Drop del dato que se desea incluir. Si al que se hace Drag&Drop es al encabezao de la tabla se anexarán todos los campos de la tabla dentro del detalle del informe.

Si el anexo del campo se hizo mediante el botón de herramientas el sistema nos llevará a un constructor de expresiones en donde se preguntará el campo a anexar.

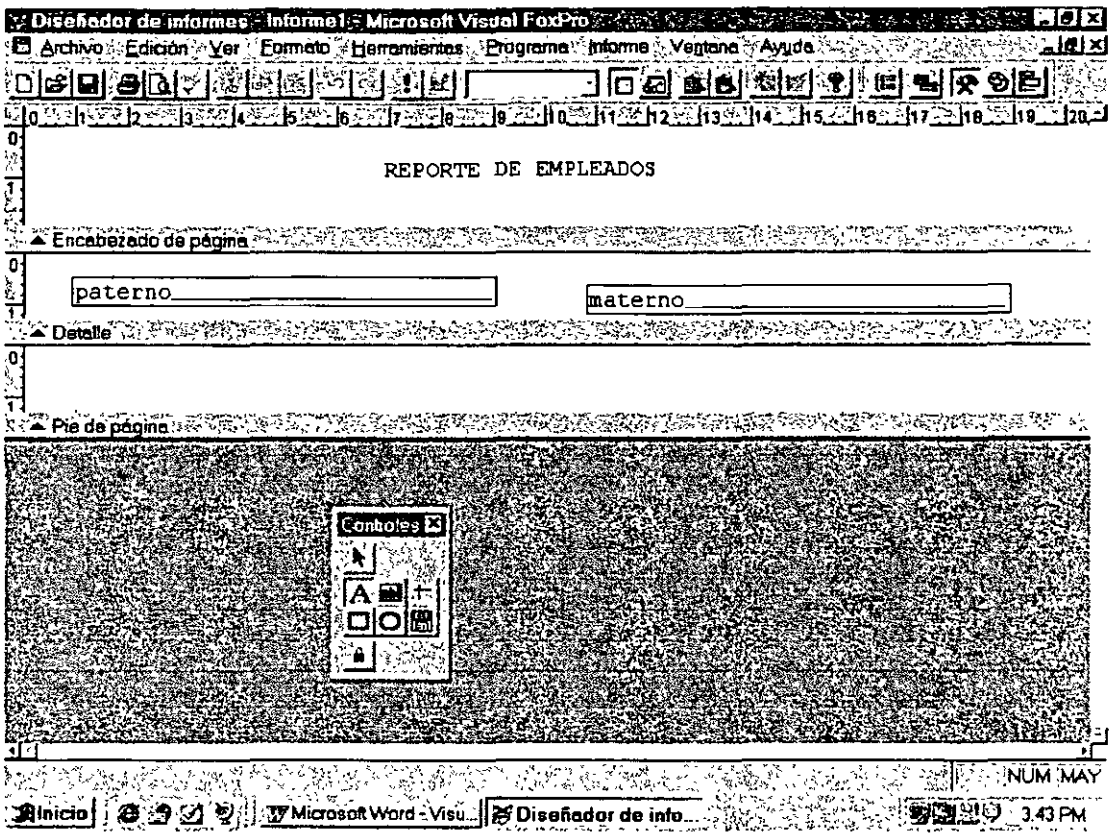
Es importante hacer notar que la referencia al campo, se hace anteponiéndole el nombre de la tabla a que pertenece y el nombre del campo, separándolos mediante un punto.



Dentro de la pantalla de construcción de expresiones distinguiremos las siguientes partes: la lista de campos de las tablas incluidas en el entorno de datos, lista de variables de memoria, operadores lógicos, matemáticos, fecha y carácter.

VII.4.2 Añadir texto

Para incluir texto dentro del reporte se deberá de oprimir el botón marcado con una A mayúscula, seleccionar a partir de donde se desea iniciar a escribir dentro del cuerpo del reporte y hacerlo, cuando se desea dejar de hacerlo se deberá de oprimir nuevamente este botón. El texto se podrá mover, para ello se deberá de seleccionar primeramente el texto por medio del click del mouse, después de esto aparecerá un marco, por medio de las flechas de desplazamiento o arrastrando el mouse será posible desplazar los campos.



En esta versión el menú de barra de la parte superior del sistema nos proporciona una opción: Formato. La cuál nos ayudará a darle el formato deseado al texto. El tipo de letra, tamaño, color, etc.

VII.4.3 Dibujo y líneas

Los tres botones con líneas, cuadro y rectángulo redondeado nos define dibujos que se podrán incluir en los reportes. De la misma manera que en los botones anteriores, se deberá de seleccionar primero el botón y después en el área del reporte marcar donde se desea que aparezca.

VII.4.4 Imágenes

Para añadir imágenes dentro de un reporte, se deberá seleccionar el botón de imagen y seleccionar el área del reporte donde se desea incrustar. Aparecerá un cuadro de diálogo donde permitirá seleccionar el archivo que se desea incluir.

VII.5 Partes del reporte

Definiremos como partes principales del reporte 3 el encabezado, el detalle y el pie de página. Estos se podrán identificar dentro del reporte porque aparecen tres barras grises membretadas con los nombres correspondientes.

Encabezado: En el área del encabezado se incluirá todo texto que se desee que aparezca al principio de cada hoja.

Detalle: Esta región servirá para incluir todos aquellos campos o variables que aparecerán en todo el reporte.

Pie de página: Aquí se definirá las variables o notas que se desea que aparezcan en el pie de todas las páginas.

Estas barras será posible desplazarlas para tener una mayor área de escritura. Mas adelante veremos que es posible que aparezcan dos barras mas que definirán los encabezados de grupo.

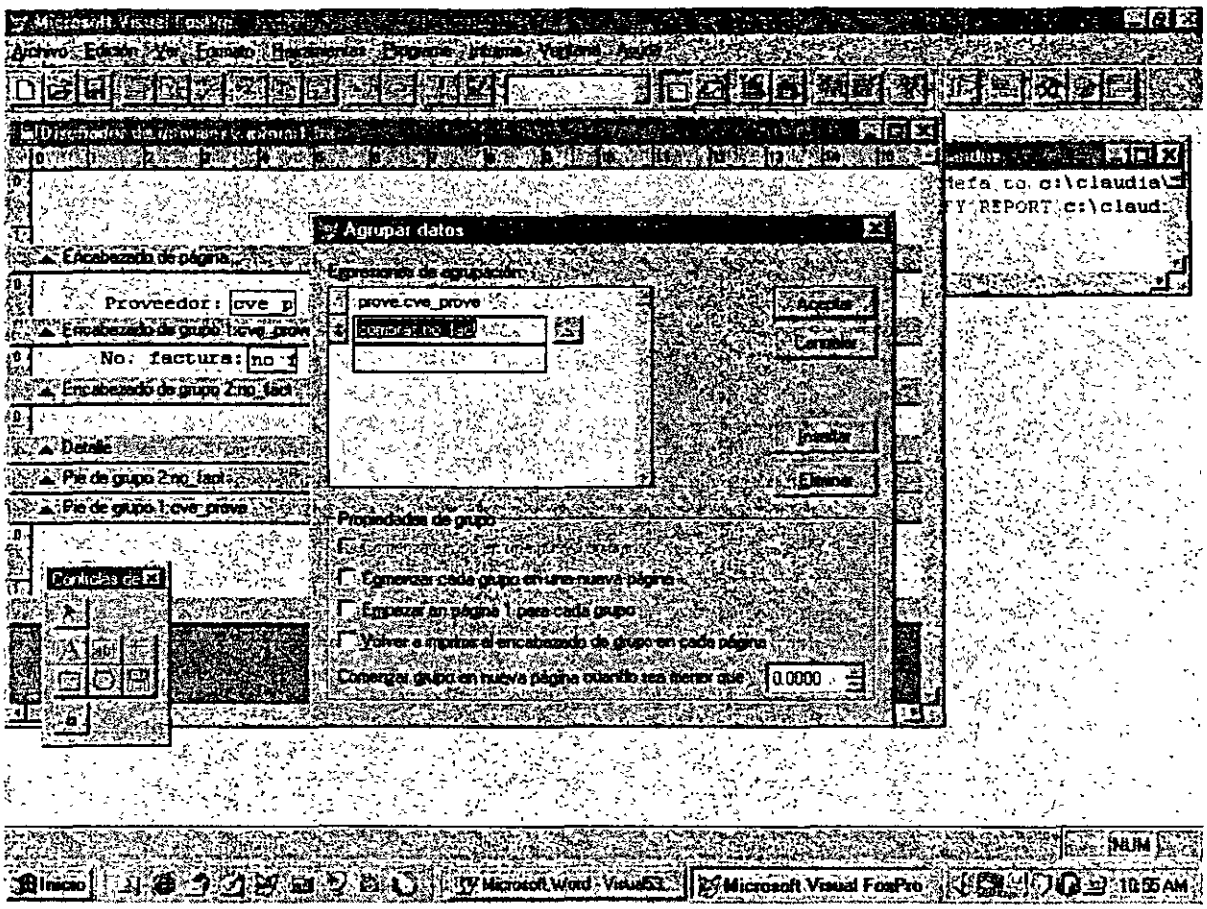
VII.6 Creación de grupos de trabajo

Para crear grupos de trabajo se deberá llamar al menú desplegable Informe-Agrupar datos. Aparecerá la primera caja de diálogo donde se nos pregunta si deseamos insertar o eliminar un nuevo grupo. Si no tenemos ninguno debemos crearlo.

Para el caso de crear uno nuevo oprimiremos el botón ya conocido de llamada al generador de expresiones.

Vamos a pensar que se desea hacer un informe en donde la empresa pueda visualizar el monto de la nómina que está pagando por departamento, por lo que desea agrupar a todos los empleados por departamento y poder así sumar los sueldos y prestaciones y finalmente compararlos.

Por lo que seleccionamos al campo de departamento y lo incluimos en la lista de grupos. Se nos harán preguntas como si se desea que aparezca en una nueva página, que inicie en uno la nueva página, etc. Esto dependerá del usuario.



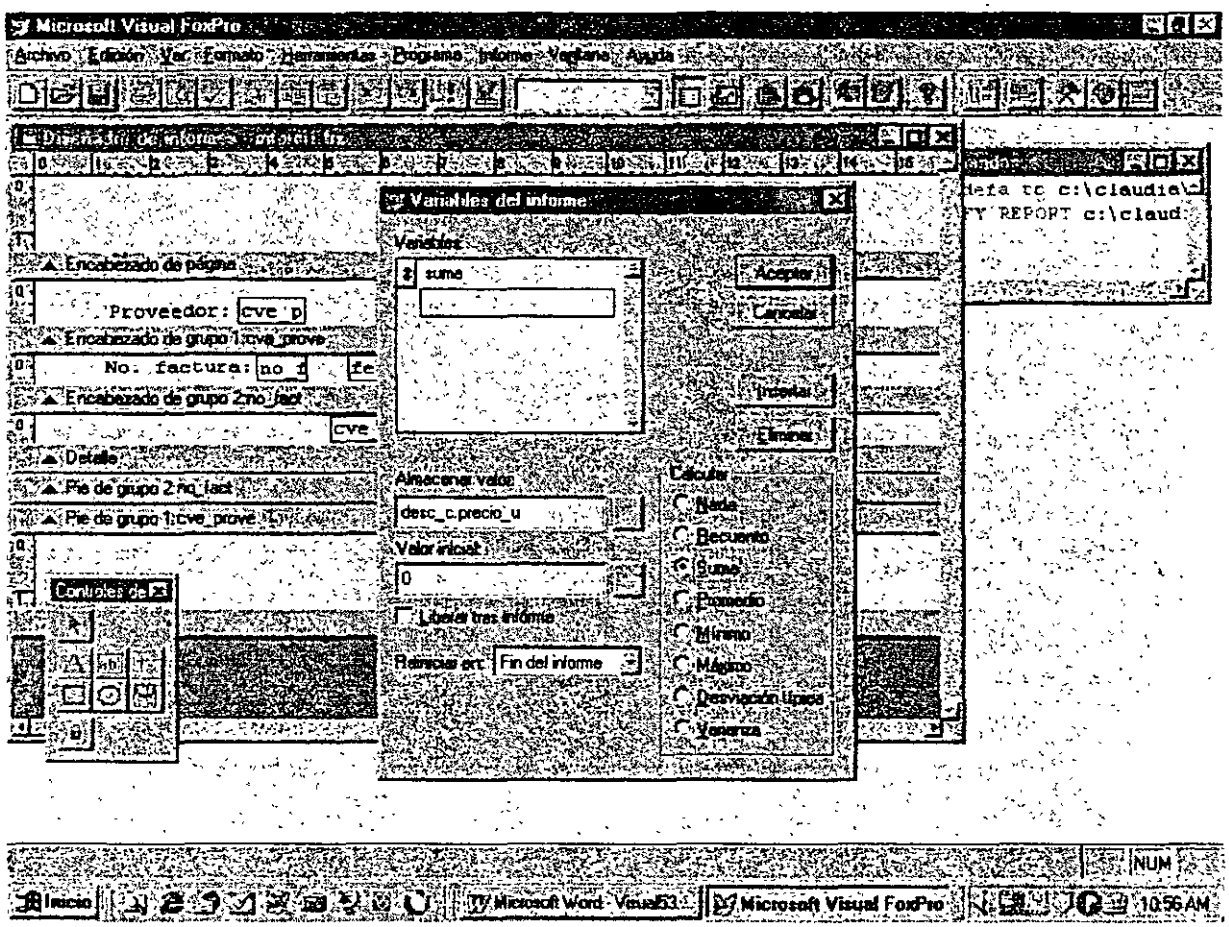
Nota: Para poder trabajar con agrupamientos será necesario abrir la base en su forma indexada utilizando como llave el campo que definiremos como campo de agrupamiento.

A partir del momento que creemos un grupo aparecerán dos bandas nuevas en el informe, en donde nos permitirá incluir aquellos datos que deseamos aparezcan como encabezados o pie de grupo.

Si nuestro objetivo es obtener sumas por departamento, en el encabezado de grupo colocaremos el nombre del departamento y en el pie las variables con las sumas.

VII.7 Creación de variables de trabajo.

Las variables de trabajo serán campos que el usuario definirá, los cuáles almacenarán la información que el usuario requiera, su naturaleza es meramente matemática por lo que almacenará cálculos de variables numéricas.



Se seleccionará del menú Informe-Variables, apareciendo una caja de diálogo en la que se nos solicita se indique cuál variable se desea insertar o eliminar. Crearemos una nueva, la pantalla nos solicitará la función aritmética que realizará, se seleccionará una de ellas, el valor de inicio, su nombre y cada cuando se reinicializará.

El botón de Valor a almacenar llamará al generador de expresiones en donde le indicaremos sobre que campo deberá de operar.

Una vez que se ha hecho toda le definición se deberá de añadir al reporte la variable creada, esto se hará en mediante el botón de cargar campos, sólo que en este caso la variable definida aparecerá en la lista del lado izquierdo (Variables). El procedimiento será el mismo que en caso definido anteriormente.

VII.8 Ejecución del reporte

Para ejecutar el reporte una vez creado, se podrá ejecutar desde la opción correspondiente en el menú de informe o por medio de la ventana de comandos y la instrucción.

REPORT FORM <NOMBRE REPORTE> <TIPO DE SALIDA>
TIPO DE SALIDA: TO SCREEN, TO PRINT, TO FILE

Reporte elaborado por código

Muchas veces un reporteador no satisface todas las necesidades del usuario, por lo que será necesario crearlo mediante código.
Cuando un reporte es creado de esta forma será necesario enviar el trabajo al dispositivo PRN (salida a la impresora) y liberar posteriormente el SPOOLER (Cola lógica de impresión), de lo contrario hasta que llegue un nuevo trabajo a la cola el primero será liberado.
Las instrucciones será las siguientes:

```
set printer to prn                  ** se define el dispositivo de salida el PRN
set printer on                      ** se enciende la impresora
SET DEVICE TO printer               ** se define como dispositivo a la impresora
```

Para liberar el trabajo haremos lo siguiente:

```
SET DEVICE TO SCREEN                ** Se regresa el control a la pantalla
set printer to c:\windows\spooler   ** Liberamos el Spooler
set printer off                     ** Apagamos la impresora
```

Reporte con variables generadas desde programación

Desde un archivo de código (PRG) se realizan todos los procesos para la impresión sólo de resultados, definiendo a todas las variables utilizadas como públicas. Por otra parte se crea el informe haciendo referencia a las variables anteriores, sin incluir a ninguna tabla en el entorno de datos y desde el programa se llamará a ejecutar al informe.

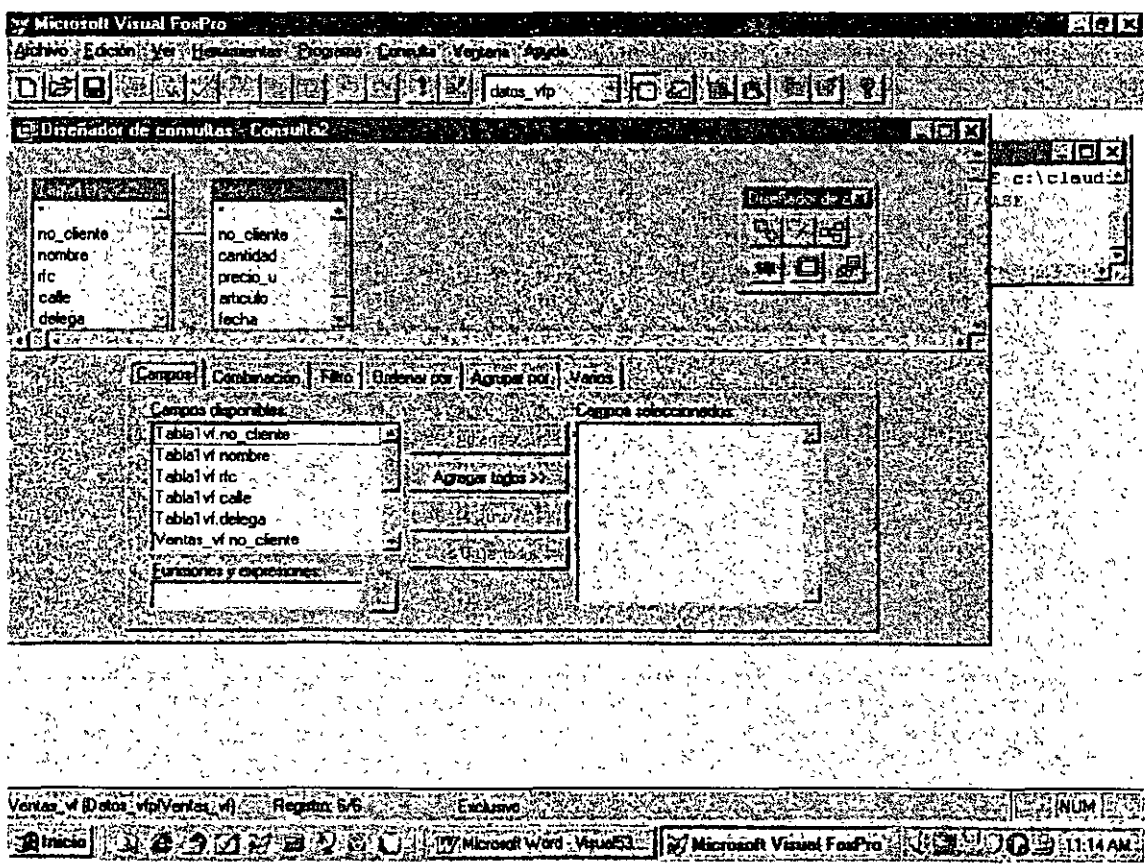
CAPÍTULO VIII. Consultas

VIII.1 Definición

Llamamos consulta al resultado de la unión entre tabals relacionadas. (Query)
Para poder realizar una consulta será necesario que las tablas que entrarán en dicha consulta posean un campo que los relacione y que exista por lo menos un datos común en ellas.

VIII.2 Agregar Tablas

El primer trabajo consiste en agragar las tablas que se utilizarán para la consulta.

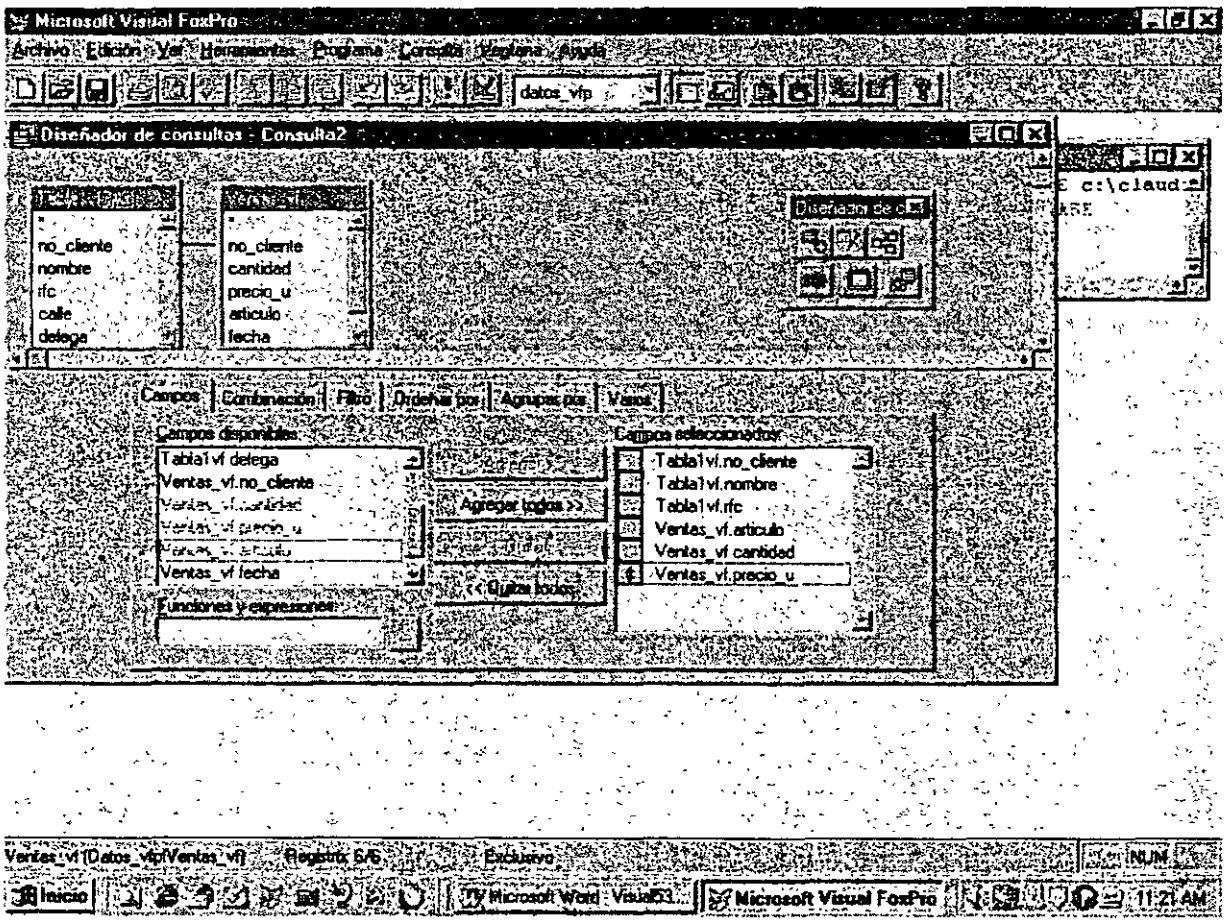


Como podemos observar en el momento de agregar las tablas mostrará la vista de éstas y sus relaciones.

En la parte inferior de la pantalla tenemos un folder con 6 pestañas, las cuáles se describirán a continuación.

VIII.3 Campos

Mostrará en una lista los campos existentes en las tablas incluídas en la consulta, de ahí seleccionaremos los que se utilizarán como salida de ésta.



VIII.4 Combinación

Es esta la parte mas importante, ya que aquí es donde le definiremos por medio de que campos se hará la relación para la generación de la consulta.
Por default mostrará los campos involucrados en las relaciones entre tablas definidas por el usuario.

VIII.4.1 Comparaciones

Esto es la condición para la comparación de los datos índices para la relación.
Se tienen cuatro diferentes tipos de comparaciones

- Inner (Interna): Especifica que sólo se incluyen en el resultado los registros que coincidan con la condición de combinación. Este tipo es el predeterminado y el tipo de combinación más comúnmente utilizado.
- Right (Derecha): Especifica que se incluyen en el resultado los registros que coinciden con la condición de combinación y los registros de la tabla a la derecha de la condición de combinación que no coinciden con ella.
- Left (Izquierda) Especifica que se incluyen en el resultado los registros que coinciden con la condición de combinación y los registros de la tabla a la izquierda de la condición de combinación que no coinciden con ella.
- Full (Completa): Especifica que se incluyen en el resultado los registros que coinciden con la condición de combinación y los registros que no coinciden con ella. El campo debe coincidir con el texto de ejemplo, carácter por carácter.

VIII.4.2 Criterios

Criterios Especifica el tipo de comparación. Los tipos de comparación son:

- Equal: Especifica que los campos tienen el mismo valor.
- Like: Especifica que el campo debe incluir caracteres que coincidan con los caracteres del texto.
- Not Like: Especifica que el campo no debe incluir los caracteres del texto.
- Exactly Like(==): Especifica que el campo debe coincidir con el texto de ejemplo, carácter a carácter.
- Not Exactly Like(Not ==): Especifica que el campo no debe coincidir con el texto de ejemplo, carácter a carácter.
- Greater Than (>): Especifica que el campo debe ser lo mismo o más que el valor del texto de ejemplo.
- Greater Than or Equal To (>=) Especifica que el campo debe ser más que el valor del texto de ejemplo.
- Less Than (<) Especifica que el campo debe ser lo mismo o menos que el valor del texto de ejemplo.
- Less Than or Equal To (<=) Especifica que el campo debe ser menos que el valor del texto de ejemplo.
- Is NULL Especifica que el campo debe contener un valor nulo.
- Is Not NULL Especifica no debe contener un valor nulo.
- Between Especifica que el campo debe ser mayor o igual que el valor inferior y menor o igual que el valor superior del texto de ejemplo. Los dos valores del texto de ejemplo se delimitan con comas. (Facturas.ifecha Between 05/10/92,05/12/92 coincidiría con registros del 10, 11 y 12 de mayo de 1992).

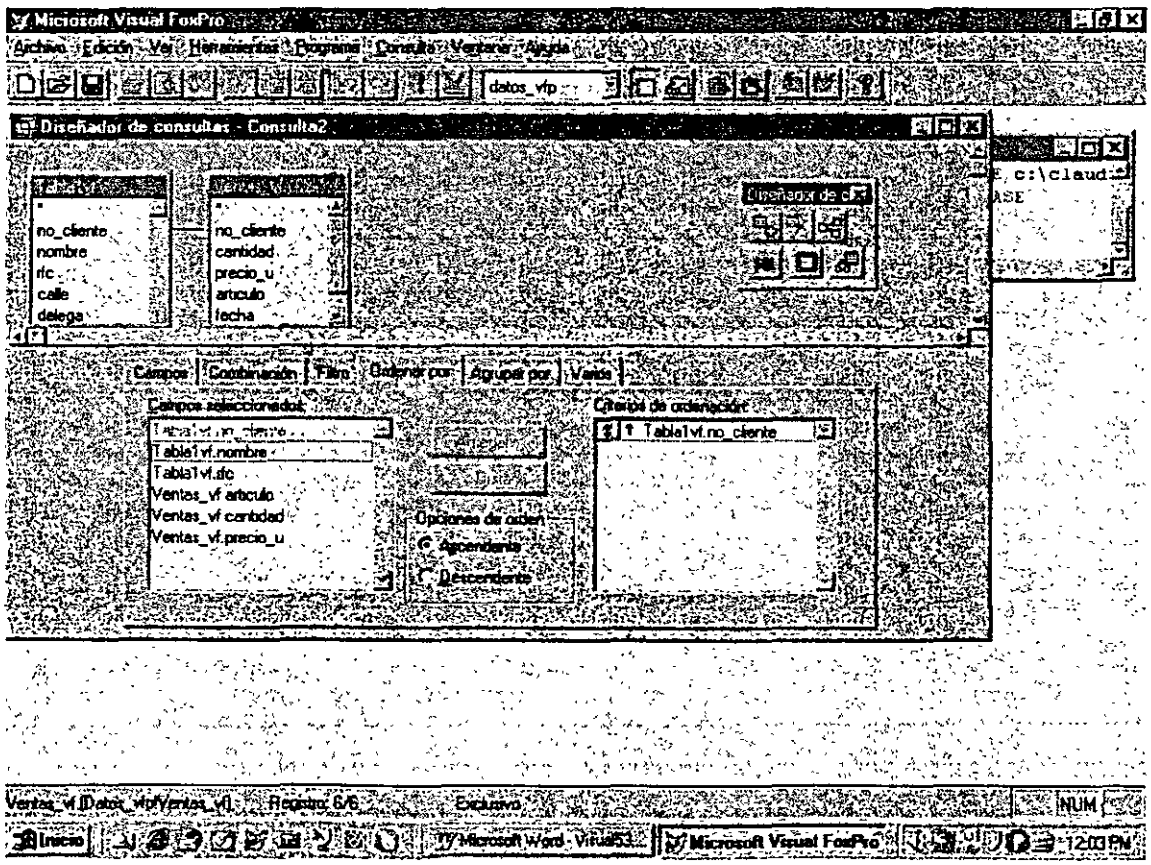
- Not Between Especifica que el campo no debe ser mayor o igual que el valor inferior y no debe ser menor o igual que el valor superior del texto de ejemplo. Los dos valores del texto de ejemplo se delimitan con comas. (Facturas.ifecha Not Between 05/10/92,05/12/92 no coincidiría con registros del 10, 11 y 12 de mayo de 1992).
- In Especifica que el campo debe coincidir con uno de los diversos ejemplos delimitados con comas del texto de ejemplo.
- Not In: Especifica que el campo no debe coincidir con uno de los diversos ejemplos delimitados con comas del texto de ejemplo.

VIII.4.3 Filtro

Para el caso de que se desee filtrar la información que se obtendrá como resultado de la unión de las tablas.

VIII.4.4 Ordenar por

Se define un criterio o campo de ordenamiento de la información.

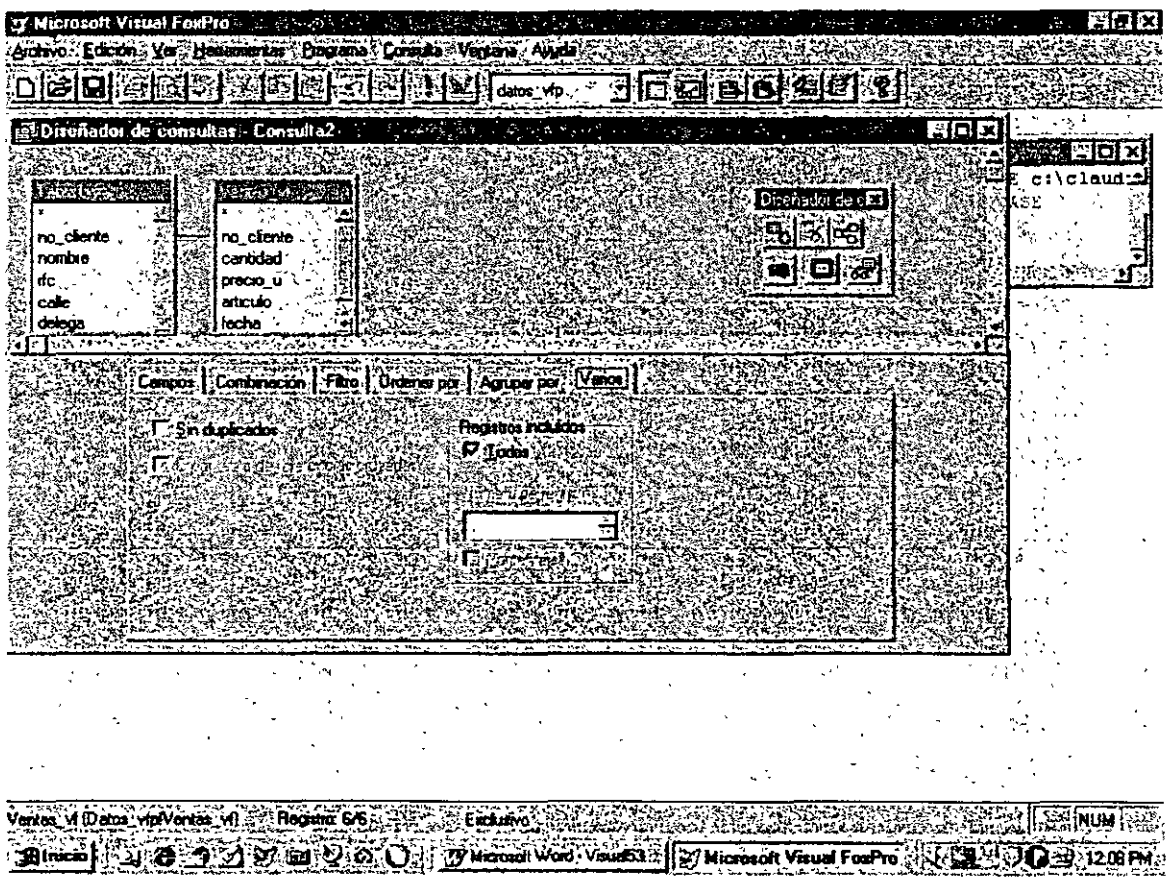


VIII.4.5 Agrupar por

Se define un criterio o campo de ordenamiento de la información.

VIII.4.6 Varios

Otros criterios como son que acepte o no duplicados y que incluya o no todos los registros.

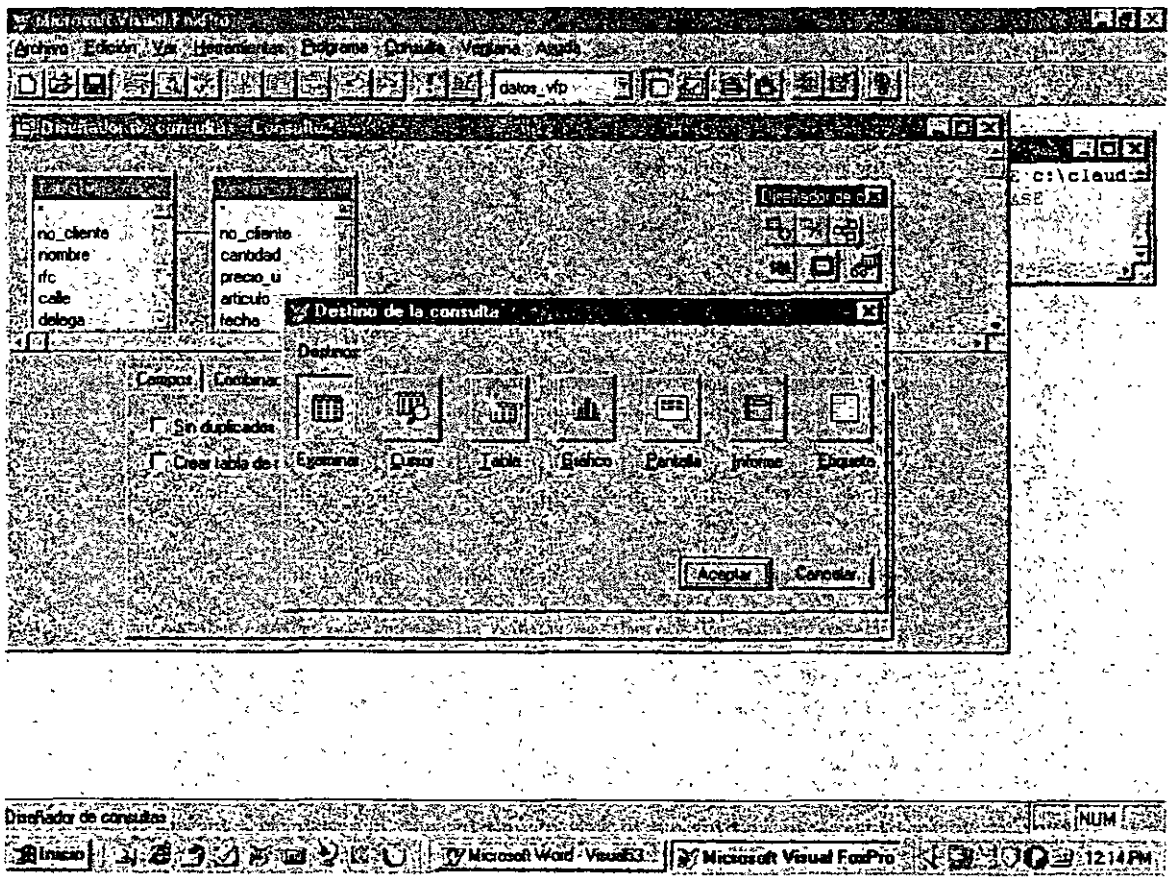


VIII.5 Salidas

Como salidas de la consulta tendremos las siguientes:

- Examinar: Muestra en un browse el resultado de la unión.
- Cursor: Tabla temporal.
- Tabla: Nueva tabla con los datos de la unión.

- Gráfico: A partir de los datos llamará al Graph de Microsoft para crear una gráfica.
- Pantalla: Se genera una Forma con la estructura resultante de los campos seleccionados.
- Informe: Se genera un Informe con la estructura resultante de los campos seleccionados.
- Etiqueta: Se generan Etiquetas con la estructura resultante de los campos seleccionados.



CAPÍTULO IX. El Ejecutable

IX.1 Definición

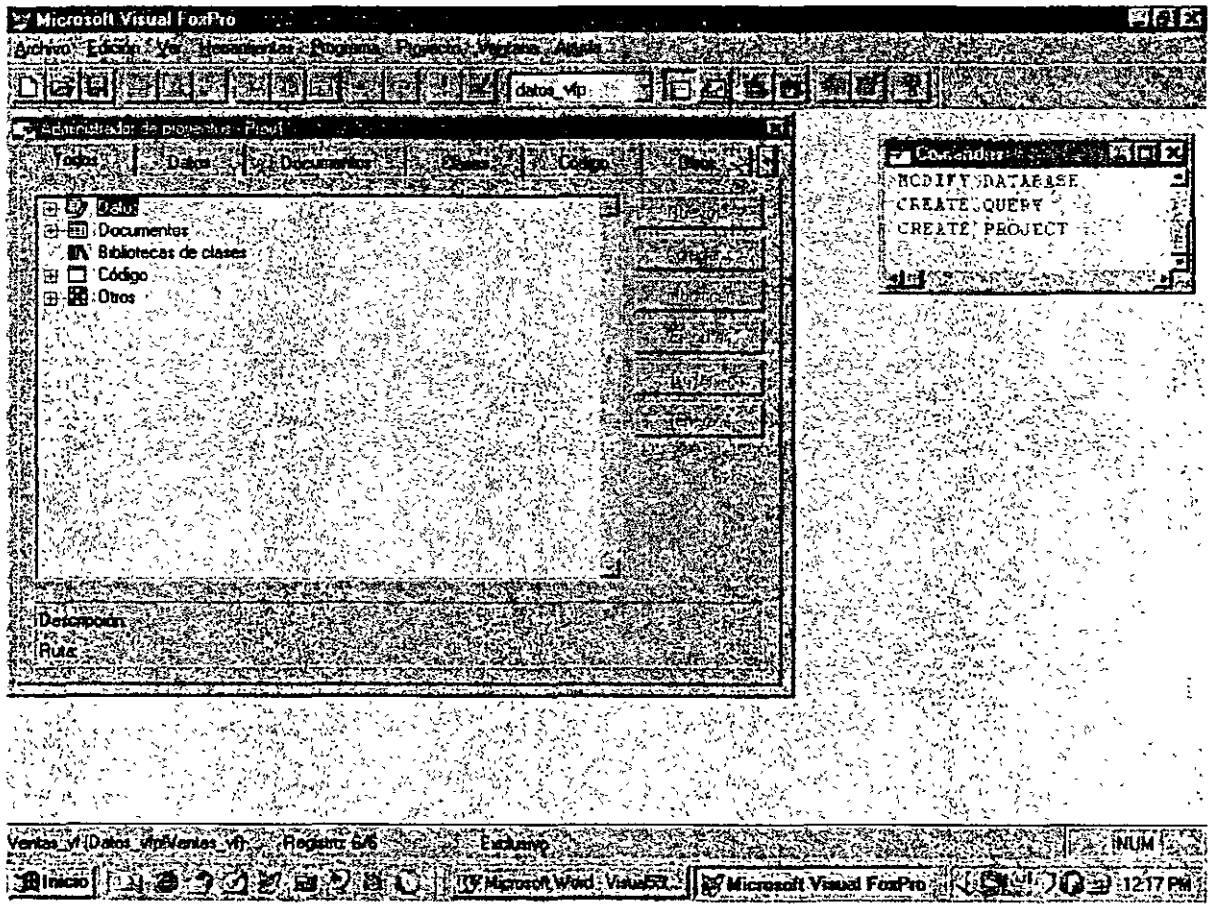
Llamaremos programa ejecutable a aquel se ejecutará con tan sólo llamarlo, seleccionándolo desde el explorador de Windows o desde un icono de acceso directo.

El archivo ejecutable se encargará de abrir tablas, ventanas, etc. e iniciar la ejecución del sistema creado.

IX.2 El Proyecto y sus Elementos

Los elementos que se incluirán en la construcción serán:
Bases de datos y tablas, formas, informes, consultas, etiquetas, programas, menús, archivos, clases.
Todos estos deberán de encontrarse dentro del proyecto. El proyecto es el que reunirá todos los elementos para poder crear un archivo ejecutable.

La pantalla que se nos muestra del administrador de proyectos es la anterior
En esta especie de folders con seis pestañas se encuentran todos los elementos que conforman al proyecto y estarán clasificados.



- Datos: Es donde se incluirán bases de datos, tablas y consultas.
- Documentos: Formas, Informes y etiquetas
- Clases
- Código: Programas, bibliotecas API y aplicaciones
- Otros: Menús, archivos de texto y otros archivos.

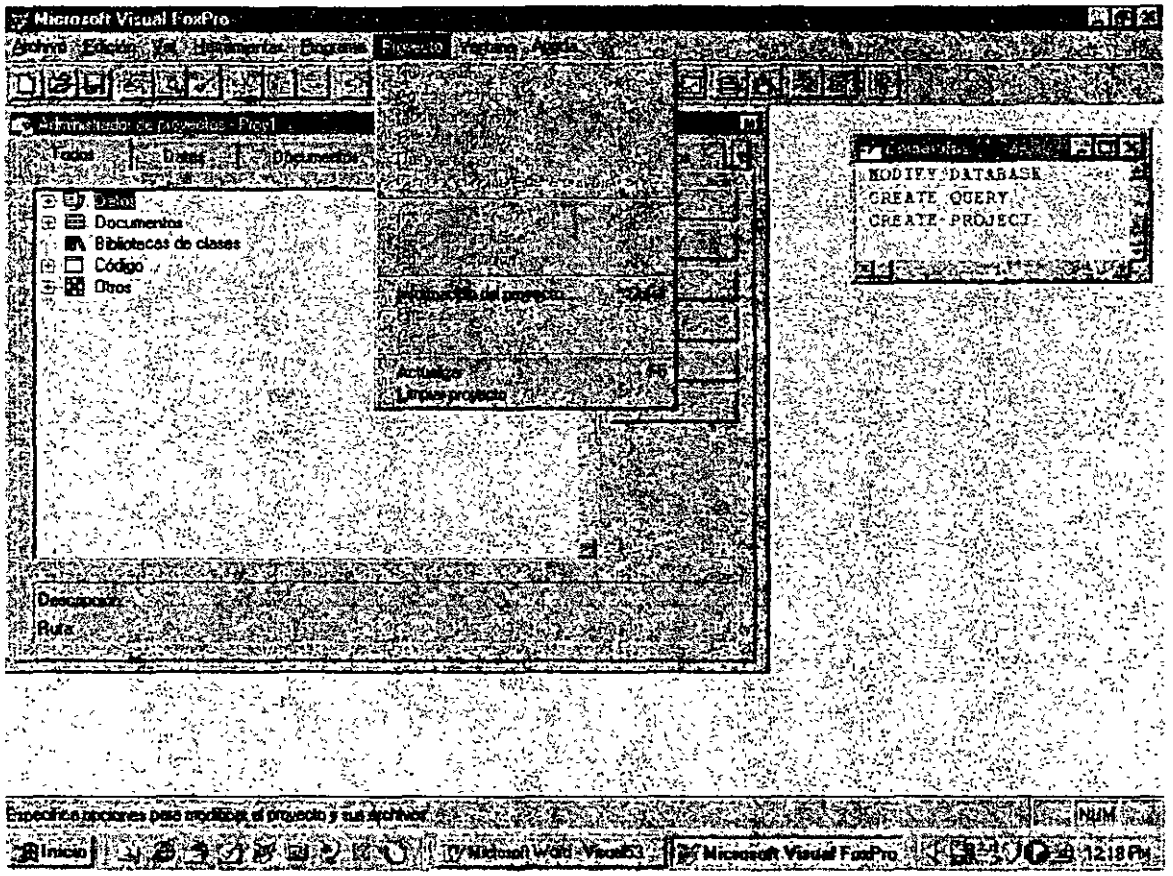
Para incluir un objeto dentro del proyecto bastará oprimir el botón de incluir y mostrará los objetos existentes del tipo solicitado, para que el usuario realice su selección.

IX.3 Menú proyecto

Visual nos muestra una nueva opción en el menú, la de Proyecto. En este encontramos opciones para agregar archivos, eliminarlos, modificarlos o cambiarles nombre.

Posee una opción de Información del proyecto en donde le escribiremos los datos del creador del sistema.

Nos permitirá definirle un icono para el momento de la instalación y mostrará los archivos que contiene el proyecto.



IX.4 Inclusión y exclusión

Llamaremos la inclusión al que un archivo que existe dentro del proyecto sea empaquetado dentro del ejecutable.

Para el caso de la exclusión serán todos aquellos archivos que estarán incluidos en el sistema pero excluidos en la construcción del ejecutable. En este caso se encuentran las bases de datos y tablas.
¿Por qué no se empaquetan? Debido a que las bases y tablas son objetos que crecen o disminuyen de tamaño, acción que no podrá ser realizada por un archivo fijo como lo será el ejecutable.

IX.5 Programa principal

Llamaremos programa principal a aquel que hará las llamadas a todos los demás elementos dentro del sistema.
Un programa principal no podrá ser una forma, deberá ser un programa (*.prg) fuente.

El siguiente es un ejemplo del código que puede conformar un programa principal.

**** Definición de variables de ambiente**

```
set sysmenu to
set century on
set notify off
set exclusive off
```

**** Se establece el nombre de la ventana principal**

```
MODIFY WINDOW SCREEN TITLE " Sistema Integral Ayuntamiento Municipal de Tehuacán, Puebla"
```

**** Se hace zoom de la pantalla**

```
zoom window screen max
```

**** Llamada a la forma principal**

```
do form USERS
```

**** Permitirá que el control permanezca dentro del sistema**

```
read events
```