



FACULTAD DE INGENIERÍA UNAM  
DIVISIÓN DE EDUCACIÓN CONTINUA

## CURSOS INSTITUCIONALES

# POWERBUILDER

Del 25 de Noviembre al 06 de Diciembre del 2002

## *APUNTES GENERALES*

CI-513

Instructora: Lic. Maricela Bustos Galicia  
DELEGACIÓN COYOACÁN  
NOVIEMBRE DEL 2002

# **Curso de PowerBuilder**

**Instructora:** Maricela Bustos Galicia  
maricela\_bustos\_galicia@hotmail.com

---

## **Manual básico**

---

- 1. Presentación**
  - 2. Descripción de área de trabajo**
    - a. Áreas de trabajo**
    - b. Barra de Herramientas en PowerBuilder**
    - c. Abriendo aplicaciones**
    - d. Ventanas**
    - e. PowerScript**
    - f. Otras Herramientas**
    - g. Base de Datos**
    - h. DataWindows**
    - i. Presentación**
    - j. Presentación Preliminar**
    - k. SQL Incrustado**
    - l. Proyecto**
  - 3. Algunos ejemplos**
  - 4. Trucos**
  - 5. Prácticas**
- Bibliografía**
- Sitios de ayuda**

## 1. PRESENTACIÓN

Este manual se compone de tres partes de una Manual básico, que describe las características generales de PowerBuilder y una sección de prácticas que contiene algunos ejercicios que ayudarán a comprender la utilidad de esta aplicación y sus herramientas.

PowerBuilder es una herramienta de desarrollo de aplicaciones para el ambiente Windows, en donde de igual forma que en cualquier programa de este tipo, la etapa de diseño consiste en escribir el código fuente del programa ejecutable.

Antes de comenzar repasaremos algunos conceptos fundamentales acerca de la programación orientada a objetos y otros términos asociados al uso de este programa. La versión que revisaremos será la 7.0; a lo largo de este manual encontraremos la descripción del ambiente de trabajo con algunos ejemplos de su uso y al final anexaremos una sección de prácticas a realizar durante las sesiones de este curso.

### Programación Orientada al Objeto

Este concepto es sencillo de entender si lo aclaramos con un ejemplo, una silla es un **objeto**, los elementos que la caracterizan son los **atributos** (color, altura, etc.). Todos estos atributos (características del objeto) pueden tomar valores dentro de un dominio definido por la característica en si (color no puede tomar el valor 1,2 metros, la altura no puede ser amarillo, etc.)

Ahora veamos su relación con los objetos en la programación, un objeto en programación es un elemento que posee características, pero más aún posee **métodos** (funciones) que han sido definidas para interactuar en operaciones comunes con dicho objeto. **Ej.** buscar un ítem en una lista. Una instancia particular de un objeto se convierte en una 'variable', para todos los fines de la programación, sus atributos se convierten en 'campos' de esta.

### Tipos de Objetos

Existen dos tipos de objetos:

a) **VISUALES**: Son aquellos que pueden ser vistos por el usuario en el monitor de su computador.

**Ej.** Botones, ventanas, etc.

b) **NO VISUALES**: Son aquellos que aunque no pueden ser vistos por el usuario, poseen todas las características de estos.

**Ej.** Errores, Objetos de Transacción (SQL)

En una aplicación PowerBuilder los objetos 'No visuales' son muy necesarios para que los objetos 'Visuales' interactúen con otros elementos ajenos al Sistema computacional en si.

## Atributos, Eventos, Métodos (Funciones)

Como ya se mencionó previamente un objeto esta constituido por 'Atributos'; estos lo caracterizan. Existen atributos que pueden modificarse y otros no. También existen atributos que son modificables tanto durante el diseño de la aplicación en si, como durante la ejecución, y otras que solo pueden ser modificadas durante el diseño.

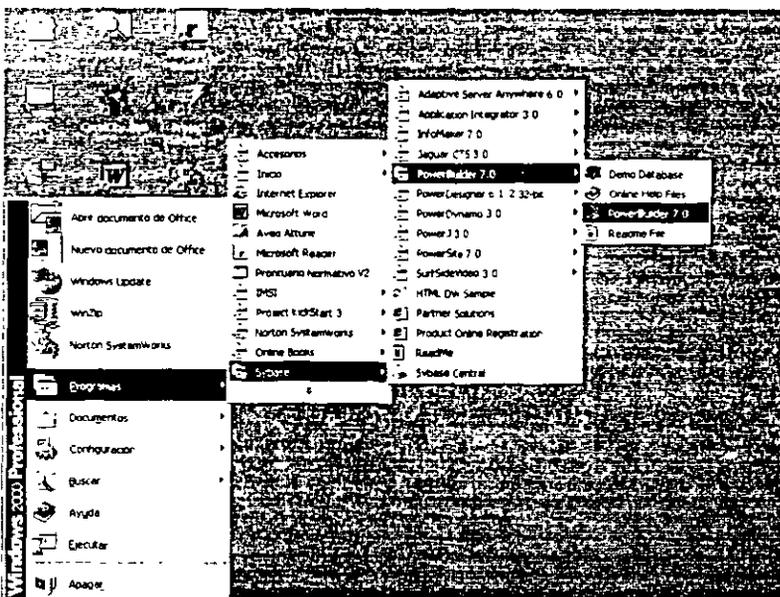
Los 'métodos', son funciones destinadas a manipular elementos que son definidos en conjunto con el objeto, esto implica que un objeto solo puede manipular elementos que estén contenidos en el (efecto "caja negra") **Ej.** en un objeto lista, la función buscar solo lo hace en los ítem del objeto lista.

Este principio se denomina **Encapsulamiento**, y permite una autonomía de cada objeto con su entorno.

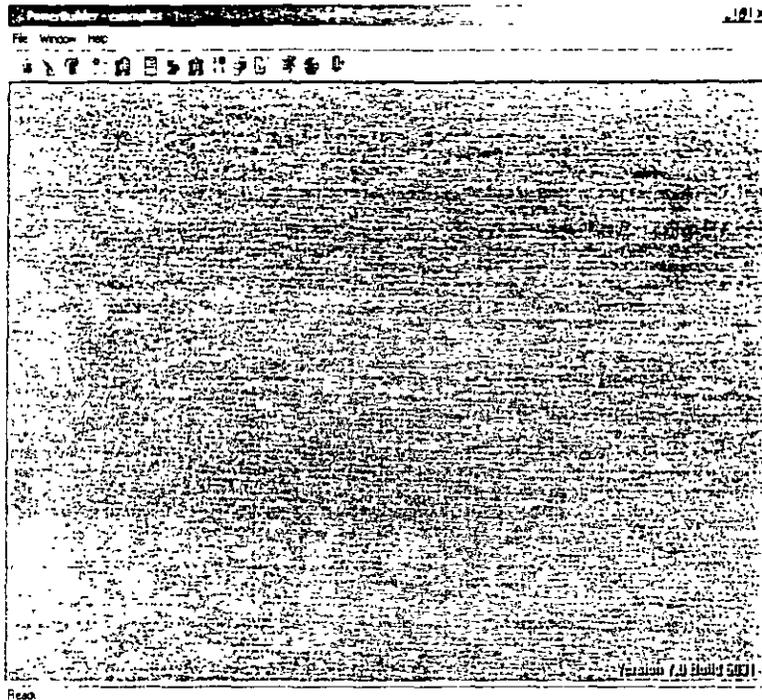
'Eventos', cada acción que se puede realizar sobre el objeto es un Evento; **Ej.** Hacer un clic sobre un objeto botón, presionar una tecla al escribir en un objeto caja de texto, etc. Cada una de estas acciones es independiente una de otras, pero no necesariamente son excluyentes. **Ej.** al presionar el botón **X**: se realizan los Eventos 'Cerrar' de la ventana, y el evento 'Destruir' también de la ventana; el primero se realiza al sacar la ventana de la pantalla y el segundo se realiza cuando se saca la ventana de la memoria principal.

También es necesario indicar el principio de la 'Herencia', como en el ser humano un hijo 'hereda' algunos rasgos de sus progenitores, en la programación orientada al objeto, un objeto puede 'heredar' los valores dados a un atributo, junto con todas las características dadas al objeto origen (código, otros objetos contenidos dentro de este, etc.); pero con la libertad de deshacerse de estos o añadir nuevos sin alterar al objeto original.

## 2. DESCRIPCIÓN DE ÁREA DE TRABAJO



## Ejemplo de secuencia para ejecutar PowerBuilder.



## Ventana de inicio de PowerBuilder.

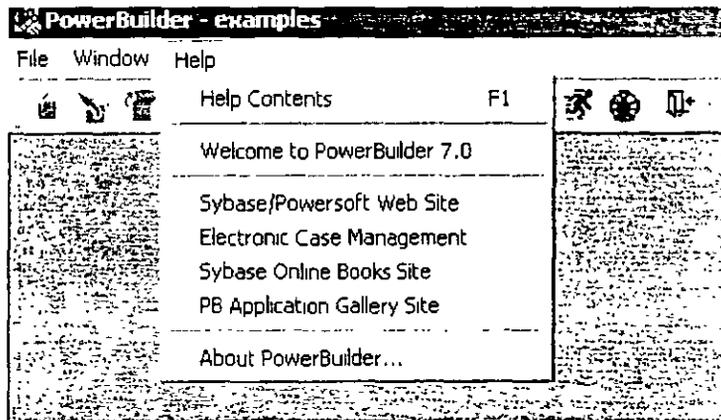
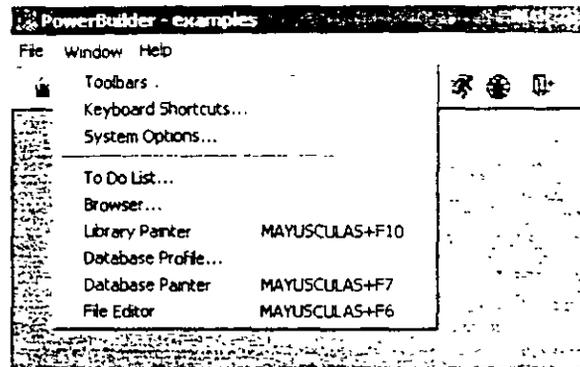
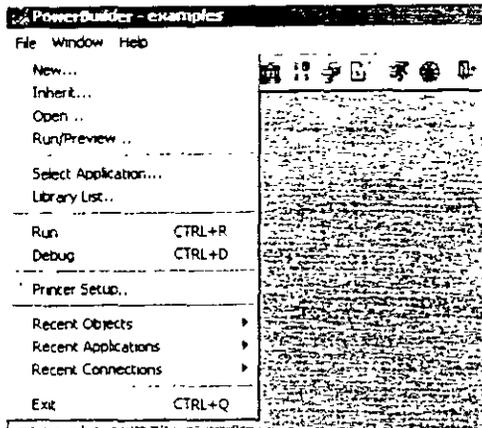
### Áreas de trabajo

El ambiente de diseño de aplicaciones de PowerBuilder esta contenido en un MDI (Múltiples Ventanas Contenidas), lo que permite que pueda soportar en forma ordenada todas las tareas que uno puede realizar en forma continua en las áreas de Desarrollo.

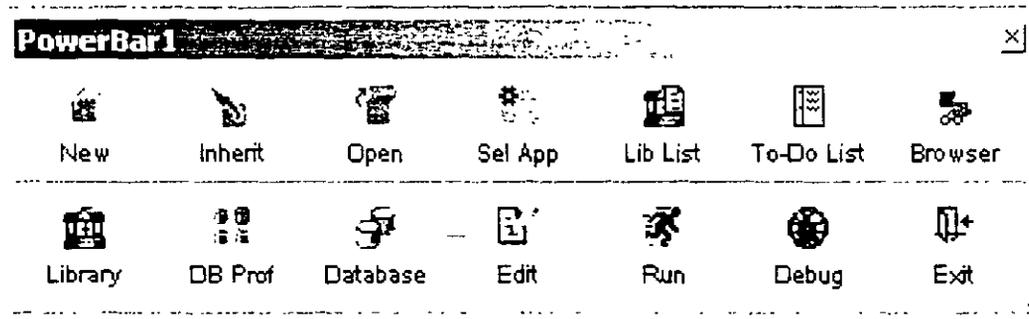
PowerBuilder posee muchas áreas de trabajo para el desarrollo de los elementos que conforman una aplicación, cada una genera un elemento que debe ser guardado en un archivo de librería (\*.pbl) donde será almacenado para su utilización. Se podrán tener abiertas múltiples áreas de desarrollo, intercambiando de una a otra a través del menú **File**, aparecerá con una marca el área donde esta actualmente y el elemento en el que se esta trabajando.

Desde el menú **Windows** se podrá definir el entorno bajo el cual trabajará el programador. Abarca desde la posición de las herramientas hasta su selección dependiendo del entorno bajo el que se esté trabajando.

El menú **Help** presenta ayudas sobre cada una de las aplicaciones contenidas el PowerBuilder, se incluyen todas estas a manera de un libro electrónico desde la guía de ayuda HTML, también podrá conectarse al sitio de Sybase para encontrar ayuda en línea, si está conectado a Internet.



### Barra de Herramientas en PowerBuilder



#### Botones de la barra de herramientas Acciones

New Crea un Nuevo objeto.

Inherit Crea una nueva ventana, utilizā objetos y menús para insertarlos en un objeto existente.

Open Abre un objeto ya existente.

Preview	Ejecuta la presentación final del proyecto en uso o presenta una vista preliminar de DataWindow.
Select Application	Selecciona una aplicación existente para trabajar o bien crea un nuevo proyecto con las herramientas de PB.
Library List	Especifica las librerías *.PBL, o bien conjunto de archivos, asociados con la aplicación actual. También pueden agregarse más documentos.
To-Do List	Realiza una lista de tareas de desarrollo que se necesitan para desarrollar la aplicación actual y crea eslabones para localizar rápidamente la sección que requiere para completar las tareas .
Browser	Busca información acerca de los objetos en la aplicación activa, de la cual se puede observar sus propiedades, funciones y variables generales, así como copiar, exportar e imprimir información.
Library	Use está área de trabajo para crear y mantener las librerías, y los objetos que contienen estas. Estos objetos pueden ser la Aplicación, Ventanas, Menús, etc. Use el botón 'Properties'  , o bien la misma opción del menú al presionar el botón derecho del ratón; para modificar la descripción dada a cada objeto. También es posible copiar, mover objetos de una librería a otra, o bien eliminarlos expresamente.
DB Profile	Permite seleccionar un profile, previamente definido, con el que se conectara a la Base de Datos correspondiente. Para cada tipo de Base de Datos los parámetros con los que se sirve para conectarse podrían variar.
Database	Permite trabajar con la Base de Datos conectada actualmente al seleccionar un profile del área DB Profile. Aquí es posible Crear/Modificar Tablas, Claves Primarias/Foráneas, Manipular datos.
Edit	Edita archivos de texto, que contiene orígenes de datos y archivos de inicialización.
Run	Permite ejecutar la aplicación desde su comienzo, si estuviera trabajando en un Área de Trabajo, se le pedirá que guarde los cambios antes de iniciar la ejecución de la aplicación.
Debug	Configura los puntos de ruptura fijos y expresiones, examina y cambia variables durante la ejecución, y observa la pila de la llamada y objetos en memoria.
Exit	Cierra la aplicación de PowerBuilder.

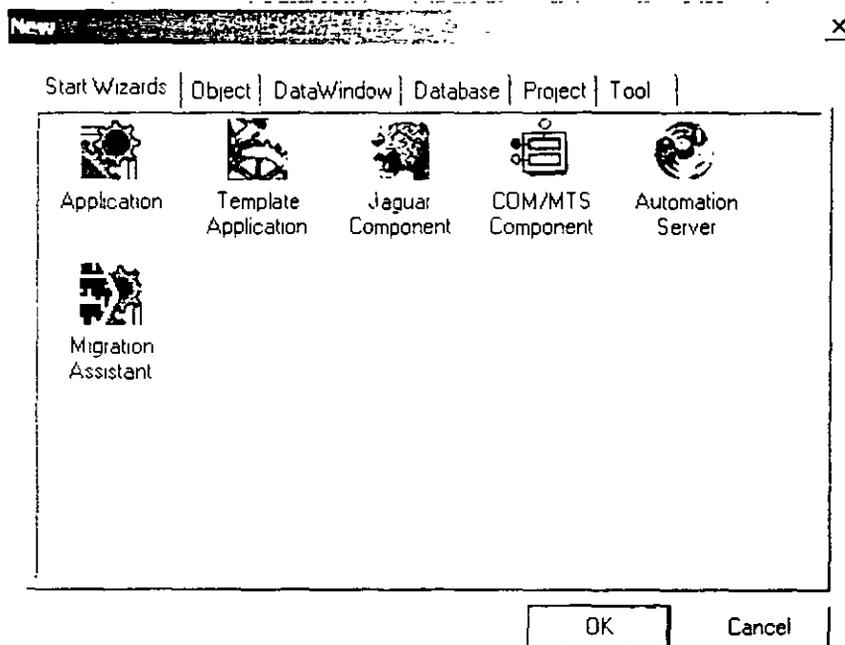
PowerBuilder usa tres barras de herramientas PowerBar, PainterBar, y StyleBar:

<b>Este barra</b>	<b>Tiene botones para:</b>	<b>Y (a menos que la oculte) se despliega:</b>
PowerBar	Abriendo ventanas y herramientas.	Siempre
PainterBar	Realiza tareas en la aplicación abierta o seleccionada.	En cada una aplicación puede existir más de un PainterBar.
StyleBar	Cambia las propiedades de texto, como conjunto de caracteres y alineación,	En aplicaciones determinadas.

### Abriendo aplicaciones

Para PowerBuilder una aplicación es un objeto, el cual debe ser único en las librerías que conformarán la aplicación en si.

Para iniciar una nueva aplicación seleccione la opción 'New' desde el ítem 'File' de la barra de menú, o bien presione el botón  desde la barra de herramientas. Aparecerá una ventana en la que deberá indicar, dependiendo de la aplicación seleccionada, las características de la aplicación en uso.



## Abrir un objeto existente

Para abrir una aplicación existente seleccione la opción 'Open' desde el ítem 'File' de la barra de menú, o bien presione el botón  desde la barra de herramientas. Busque el archivo (\*.pbl) que contenga el objeto aplicación.

## Guardar Cambios

Para guardar los cambios al modificar seleccione la opción 'Save' desde el ítem 'File' de la barra de menú, o bien presione el botón  desde la segunda barra de herramientas.

## Propiedades

Presione el botón  o vaya al menú Entry opción Properties.

Los atributos de la aplicación tienen relación con características globales como son fuente predeterminadas de las letras para las cajas de texto, etiquetas, etc. También se puede definir el icono que tendrá la aplicación cuando se haga el archivo ejecutable de ésta.

Tópico	Descripción
General	Permite obtener información del objeto application.
.. Font	Permite definir características de presentación visual de elementos.
Icon	Permite definir el icono de la aplicación.
Variable...	Permite conocer las variables globales por defecto y su tipo.
Libraries	Permite declarar las librerías que conforman la aplicación.

## Eventos

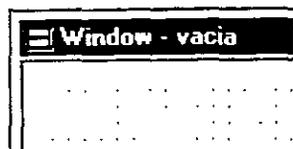
Los eventos más utilizados para un objeto 'application' son:

Evento	Descripción
Open	Las acciones cuando inicie la ejecución de la aplicación.
Close	Las acciones cuando termine la ejecución de la aplicación.

## Ventanas

Las Ventanas son como contenedores, es posible poner en ellos una diversidad de objetos. Estos objetos se denominan en forma genérica '**Controles**'.

Para trabajar en el diseño de una ventana presione el botón , del fichero Object, en la ventana de New, aparecerá una ventana donde podrá seleccionar una de las ya existentes y presionar 'OK' para seguir trabajando.



En la parte superior aparece el nombre del objeto ventana. El rectángulo que esta inmerso en la ventana corresponde al objeto ventana en diseño.

## Nueva Ventana

Después de presionar el botón 'New' aparecerá una ventana vacía. puede redimensionar desde los bordes de la misma. Al cerrar la ventana de Diseño, se le preguntara por el nombre que le dará a la nueva ventana.

## Propiedades ventana

Las propiedades de la ventana están organizados en forma de tópicos. Aquí no se mencionarán todos, ya que algunos se repiten para todos los 'Controles' (ver Controles)

Tópico	Descripción
<b>General</b>	<p>Casilla 'Title', indica el texto de titulo de la ventana. La casilla 'Menú Name', indica el nombre del objeto Menú para la ventana. 'Window Type', indica el tipo de ventana:</p> <ul style="list-style-type: none"> <li>• Main, ventana normal, con posibilidad de menú.</li> <li>• MDI Frame, ventana que posee área interna para contener otras ventanas.</li> <li>• MDI Frame with ..., igual que la anterior, pero con barra de estado inferior.</li> <li>• Identifica al Icono que aparecerá en la parte izquierda de la cabecera de la ventana.</li> </ul>
<b>Scroll</b>	Permite definir si la ventana tendrá barras de desplazamiento, y la unidad a utilizarse por ellas.
<b>ToolBar</b>	Permite definir si se utilizara la caja de herramienta definida para el menú, su presentación y posición.
<b>Other</b>	Permite indicar la posición de la ventana dentro de la pantalla, como será presentada. Utilice el esquema inferior para mover la ventana.

## Controles

Son los objetos visuales que pueden ser puestos en la ventana. En la barra de herramientas secundaria, el botón  que esta en el segundo grupo, muestra el listado de todos los objetos elementales que es posible de utilizar. Adicionalmente se pueden diseñar objetos con características adicionales para ser utilizadas también.

Los controles básicos son:

Control	Icon	Descripción
<b>CommanButton</b>		Boton, puede visualizar un texto en el. Presenta la capacidad visual de sobresalir y achatarse al presionar el ratón sobre el.
<b>PictureButton</b>		Botón con n. Tiene todas las características del control anterior, pero adicionalmente permite mostrar una n dentro del mismo botón.
<b>CheckBox</b>		Permite hacer una selección (marcarlo) entre varias alternativas. En el primero cada alternativa es independiente. En el segundo solo una alternativa, del conjunto, puede seleccionarse
<b>RadioButton</b>		
<b>StaticText</b>	<b>A</b>	Caja de texto, como etiqueta. Solo para mostrar información, no hay posibilidad que pueda escribirse directamente en el.
<b>Picture</b>		Caja de dibujo, permite visualizar una n. Esta puede ser definida en tiempo de diseño, o puede ser modificada en tiempo de ejecución.
<b>SingleLineEdit</b>		Caja de edición de texto, es posible que el usuario en tiempo de ejecución escriba un texto dentro de la misma caja.
<b>EditMask</b>		Caja de texto, a la que se le puede restringir el conjunto de caracteres a ingresar. Permite un formato de presentación.
<b>DropDownListBox</b>		Caja de lista desplegable. Muestra un solo ítem, el seleccionado, de la lista que se despliega al presiona el botón adjunto.
<b>ListBox</b>		Caja de lista. Muestra todos los ítems a la vez, pudiendo seleccionarse inclusive, más de uno.
<b>Datawindow</b>		Datawindow Control, el que se vincula con un objeto Datawindow diseñado.

Para añadir un control a la ventana, seleccione el control, del listado y arrástrelo con el ratón hacia el lugar en la ventana donde desea añadirlo. También puede seleccionar varios controles con la tecla CTRL.

## Propiedades Controles

Las propiedades de los controles pueden ser modificados, tanto en tiempo de diseño, como en tiempo de ejecución; vía código.

Para ver las propiedades de un control, presione el ratón dos veces.

Las propiedades de todo control están ordenadas por tópicos, estos pueden variar levemente entre controles. Los más comunes son:

Tópico	Descripción
<b>General</b>	<p>'Name', indica el nombre del objeto. La casilla 'Text', indica el texto a ser mostrado. 'Tag', permite guardar cualquier dato. 'Visible', indica si el objeto puede verse. 'Enabled', indica si objeto esta activamente disponible.</p> <ul style="list-style-type: none"><li>• Para los Botones: 'Default' indica que si se presiona el Enter en cualquier parte de la ventana, se activa dicho botón. 'Cancel' indica que si se presiona la tecla Escape, se activa dicho botón.</li><li>• Para las cajas de Edición, Etiqueta y Lista Desplegable: 'Alignment', indica la alineación del texto. 'Case' el tipo de presentación. 'Border' indica el tipo de borde. 'Limit', el número de caracteres a dejar ingresar.</li><li>• Para la Lista Desplegable: 'Allow Edit', indica que se puede escribir un ítem además de seleccionarlo. 'Sort', indica que el listado será ordenado alfabéticamente.</li></ul> <p>Para la Lista: 'Multi Select', indica posibilidad de selección múltiple.</p>
<b>Font</b>	Permite definir el tipo de letra, tamaño y características de presentación del texto visible. 'Text Color', indica el color del Texto. 'Background', indica el color de fondo.
<b>Position</b>	Indica las coordenadas 'X' e 'Y', también se indica el largo (Width) y el alto del objeto (Height)
<b>Icon</b>	Indica el tipo de Icono, que será utilizado por el cursor cuando este pase por encima del objeto en tiempo de ejecución.
<b>Items</b>	<ul style="list-style-type: none"><li>• Para los objetos lista, permite ingresar el texto de cada ítem.</li></ul>
<b>Mask</b>	<ul style="list-style-type: none"><li>• Para la mascara, permite definir la mascara de entrada de datos. Seleccione 'Type' para ver en 'Masks', los posibles caracteres a utilizar. Cualquier otro carácter se considera un litera.</li></ul>

## Eventos Controles

Los eventos más comunes, los que no necesariamente están presentes en todo control son:

<b>Evento</b>	<b>Descripción</b>
<b>Clicked</b>	Cuando el usuario presione el ratón sobre el área que conforma el objeto visual.
<b>GetFocus</b>	Cuando el objeto obtiene la atención, se ratón el cursor sobre el.
<b>LoseFocus</b>	Cuando el objeto pierde la atención.
<b>DoubleClicked</b>	Cuando el usuario presiona dos veces el ratón sobre el área que conforma el objeto visual.
<b>SelectionChanged</b>	Cuando se cambio de ítem seleccionado.
<b>RButtonDown</b>	Cuando se presiona el botón derecho del ratón sobre el área que conforma el objeto visual.
<b>Modified</b>	Cuando el control pierde la atención, se presiona Enter o Tab, o el texto es modificado.

## PowerScript

Todo lenguaje de programación contiene sentencias propias que le permiten manipular la información, permite separar cursos de acción dependiendo de ciertos valores, o bien sencillamente presentarla al usuario. Todo esto debe ser realizado por el programa en forma autónoma y a veces sin participación externa, por ende se debe ser precursor al momento de diseñar y programar una aplicación. El siguiente capítulo presenta las sentencias que son requeridas para la manipulación de la información por parte del programa en sí.

### Tipos de datos

Los tipos de datos que están definidos para PowerBuilder son los siguientes, se denominan "Tipos de Datos Estándar".

Tipo de Dato	Descripción
<b>Blob</b>	Objeto binario largo. Usado para almacenar una gran cantidad de datos, pero cuyo largo no está claro. (Ej. Imágenes, archivos de texto, documentos)
<b>Boolean</b>	Contiene solo Verdadero o Falso. (TRUE o FALSE)
<b>Char</b>	Almacena un único carácter ASCII.
<b>Date</b>	Incluye una fecha, en formato americano (yy[yy],mm, dd) Año (1000-3000), mes (01- 12) y días. (01- 31)
<b>DateTime</b>	Contiene la Fecha y la Hora en un único tipo de dato, usado solo para lectura o escritura de valores 'DateTime' definidos en una Base de Datos. Para convertir un valor 'DateTime' en un tipo de dato que pueda ser usado en PowerBuilder (date, time), use:  Date(DateTime), para convertir un valor 'DateTime' en un valor Date.  Time(DateTime), para convertir un valor 'DateTime' en un valor Time.  DateTime(Date, Time), para convertir un 'Date' y un 'Time' en un valor DateTime
<b>Dec</b>	Soporta números decimales con sobre 18 dígitos. Ej. 123456,000001
<b>Double</b>	Números con punto flotante, con 15 dígitos de precisión y un rango desde 2.2E308 a 1.7E+308.
<b>Int</b>	Un entero de 16 bits, con rango de -32768 a +32767.
<b>Long</b>	Un entero largo de 32 bits, con rango de -2,147,483,648 a +2,147,483,647.
<b>Real</b>	Un número con punto flotante, con 6 dígitos de precisión y un rango desde 1.17 E -38 a 3.4 E +38.
<b>String</b>	Cualquier carácter ASCII con un largo variable. (0 a 60,000)
<b>Time</b>	Hora en formato 24 horas, incluye la hora (00-23), minutos (00- 59)

	segundos (00- 59) y fracción de segundos (sobre 6 dígitos). Con un rango desde 00:00:00 a 23:59:59.999999
--	---

### **Definición de Variables/Constante**

Una variable se define anteponiéndole el tipo de dato que sera permitido para esa variable.

**Ej.** Se desea declarar una variable de tipo entera y de nombre 'monto\_cuota'

```
integer monto_cuota
```

```
string nombre
```

puede declarar varias variables del mismo tipo, separando cada una de estas con una coma (,)

Ej.

```
integer monto_cuota, total_cuota= 0
```

Nota:

Como podrá notar a la variable se le asigno un valor inicial. Esto se puede realizar con cualquier variable, teniendo cuidado de asignar un valor que corresponda con el tipo de dato.

Para declarar una variable como una constante, el valor asignado inicialmente no podrá ser modificado durante el programa, se antepone la palabra clave CONSTANT.

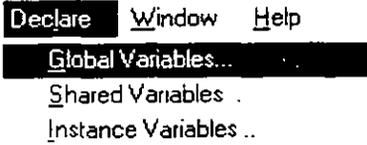
Ej.

```
constant real LR_PI = 3.14159265
```

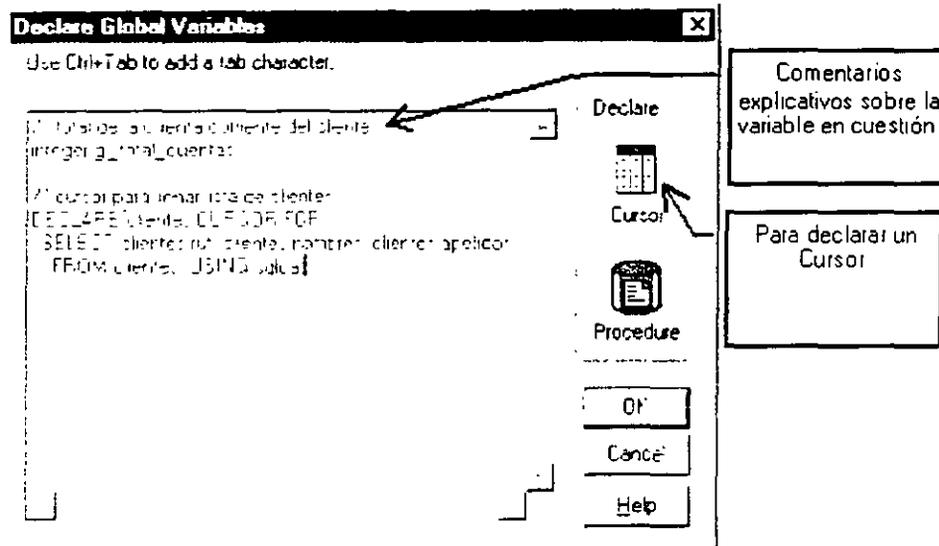
### **Ámbito**

Existen 4 tipos de ámbito, dentro de los módulos del programa, en donde podrá ser utilizada una variable. Estos cuatro ámbitos serán mencionados a continuación.

Los tres primeros son definidos fuera de cualquier modulo ejecutable, y se declaran en el Área de trabajo de las Ventanas, en el menú 'Declare'.



Todos estos tienen la misma ventana de ingreso de variables, solo cambiando el ámbito (rango) de uso de la variable dentro del programa.



## Global

Es aquella variable/constante que podrá ser utilizada en toda la aplicación, es decir; podrá hacerse referencia a ella desde cualquier módulo del programa.

Para declarar una variable global. Estando en cualquier módulo de código, seleccione el menú 'Declare' opción 'Declare Global'; aparecerá la ventana de declaración de variables, donde podrá definir la(s) variable(s) como se explico anteriormente.

También puede declarar un 'cursor' (vea el tópico SQL Incrustado) que podrá ser referenciado desde cualquier parte de la aplicación. Para ello solo presione el dibujo que dice 'Cursor' y siga las mismas indicaciones que se dan en la Sección 'Cursores' del tópico 'SQL Incrustado'.

### Globales solo al Objeto (Instance)

Las variables que sean declaradas como Globales solo al Objeto, podrán ser utilizadas en cualquier código fuente, pero

solo del objeto donde se declaro (ventana, menú, objetos del usuario), es decir, donde se define la(s) variable(s). También podrán ser utilizadas por todos los objetos que estén inmersos en el (Ej. botón en una ventana)

Para declarar una variable global al Objeto. Estando en cualquier modulo de código o en la definición de la ventana, seleccione el menú 'Declare' opción 'Declare Instance'. Aparecerá la ventana de declaración de variables, ventana donde podrá definir la(s) variable(s) como se explico anteriormente.

También puede declarar un 'cursor' (vea el tópico SQL Incrustado) que podrá ser referenciado desde cualquier parte de la ventana especifica. Presione el dibujo que dice 'Cursor' y siga las mismas indicaciones que se dan en la Sección 'Cursores' del tópico 'SQL Incrustado'.

### **Recordantes (Shared)**

Las variables declaradas de este tipo igual que las del tipo anterior (Instance) solo pueden ser reconocidas dentro del objeto especifico donde se declaro (ventana, menú, objeto del usuario) , pero estas últimas tienen la particularidad de *conservar* el valor almacenado en esta al momento de cerrar y abrir nuevamente el objeto en cuestión.

Para declarar una variable global al Objeto de tipo Shared. Estando en cualquier modulo de código o en la definición de la ventana, seleccione el menú 'Declare' opción 'Declare Shared'. Aparecerá la ventana de declaración de variables, ventana donde podrá definir la(s) variable(s) como se explico anteriormente.

También puede declarar un 'cursor' (vea el tópico SQL Incrustado) que podrá ser referenciado desde cualquier parte de la ventana especifica. Presione el dibujo que dice 'Cursor' y siga las mismas indicaciones que se dan en la Sección 'Cursores' del tópico 'SQL Incrustado'.

### **Locales (al Modulo)**

Estas variables son declaradas en el mismo código del modulo en cuestión, y solo tienen vida en este, no pueden ser compartida por otros módulos del mismo objeto inclusive.

```

Script - open for [Untitled] returns long
Select Event | Paste Argument | Paste Object
//total de cuenta corriente del cliente
integer li_total_cuenta

//cursor para llenar lista usuario
DECLARE clientes CURSOR FOR
SELECT cliente.rut, cliente.nombre, cliente.apellidos
FROM clientes USING sqlca;

```

### Utilización

Una variable puede ser utilizada para asignarle un valor, comprobar su valor, asignarle su valor a otra variable **Ej.** Asignarle un valor.

**monto\_cuota=13**

**nombre= 'Ricardo'**

Comprobar su valor

If **monto\_cuota=13** Then ...

If **nombre<> 'Ricardo'** Then ...

Asignarle su valor a otra variable

**total\_cuota= monto\_cuota**

Una variable también puede aparecer tanto en la parte 'asignante ' (quien recibe el valor) como en la 'asignadora' (quien genera el valor a asignar)

Ej.

**monto\_cuota=monto\_cuota+12**

### Operadores Matemáticos, relacionales y lógicos

Los operadores matemáticos son los siguientes:

Suma	+
Resta	-
Multiplicación	*
División	/

Además existen los agrupadores de expresiones matemáticas:

'(', indica comienzo de la Agrupación.

)', Indica fin de la agrupación

Ej.

**monto\_cuota = (23 \* monto\_cuota)\* 13**

Los operadores relacionales son los siguientes:

O inclusivo	OR
y inclusivo	AND
Negación	NOT
Igualdad	=
Desigualdad	<>
Menor	<
Menor o igual	<=
Mayor	>
Mayor o igual	>=

Ej.

If **monto\_cuota=12 AND NOT objeto.enabled** Then ...

Adicionalmente existen 2 valores constantes.

Verdadero	True
Falso	False

## Flujos de Control

Los flujos de control permiten que las acciones ha ser realizadas puedan ser bifurcadas a otra dependiendo de alguna expresión condicional. Existen 2 tipos de flujos de control

- Condicional
- Iterativo

### Condicional

Se refiere a que el flujo de control es bifurcado desde una sentencia, o conjunto de estas, a otra, o conjunto de estas. Existen 2 clase de Flujos de control condicionales

## IF..THEN..ELSE

La sentencia 'If' permite bifurcar la ejecución del código entre 2 caminos posibles dependiendo de la veracidad o no de una expresión condicional

El tamaño, en líneas de código, de los segmentos, 'verdadero' o 'falso' no tienen un límite.

```
Ej.  If monto_cuota=13 Then
      nombre= 'Ramiro'
      Else      nombre=
'Francisco'
      End If
```

También es posible anidar varios 'If..Then..Else' consecutivamente.

```
Ej.
If monto_cuota=13 Then
nombre= 'Ramiro'
ElseIf monto_cuota=15 Then
nombre= 'Roberto'
Else
nombre= 'Francisco'
End If
```

## CHOOSE

Permite bifurcar la ejecución del código entre múltiples caminos posibles; dependiendo de la veracidad o no de una expresión condicional, principalmente una variable.

El tamaño, en líneas de código, de los segmentos 1..'n' no tienen un límite.

**Ej.**

Choose case **monto\_cuota**

Case Is< 13: nombre='Ramiro'

Case 16 To 48: nombre= 'Roberto'

Else

nombre='Francisco'

End Choose

Cada Case <expresión> puede tener los siguientes valores:

- Un valor
- Una lista de valores separados por coma. (**Ej.** 2, 4, 6, 8)
- Una cláusula TO, desde hasta. (**Ej.** 1 TO 30)
- Una cláusula IS seguida por un operador relacional y un valor a comparar. (**Ej.** IS>5)
- Cualquier combinación de lo anterior, implícitamente indica un OR. (ej.1, 3, 5, 7, 9, 27 TO 33, IS >42)

### **Iterativo**

Se refiere a la repetición de una(s) sentencia(s) que son definidas dentro de un bloque clarificado por una sentencia iterativa. Existen 2 clases de Flujos de control Iterativos.

### **FOR..NEXT..STEP**

La sentencia 'for' permite repetir una(s) sentencia(s) un número definido anterior al comienzo de la ejecución de estas sentencias.

El tamaño, en líneas de código, del segmento 1, no tienen un límite.

Ej.

```
for i= 1 to 23
```

```
  j= i + (j * 1.2 )
```

```
next
```

La variable 'i' se denomina "contador", el valor final de iteraciones no debe ser superior al rango definido para el tipo de dato del contador. Si el valor de 'i' es superior al valor final de iteración, las sentencias no se ejecutan.

Ej.

```
For n = 5 to 25 step 5
```

```
  a = a+10
```

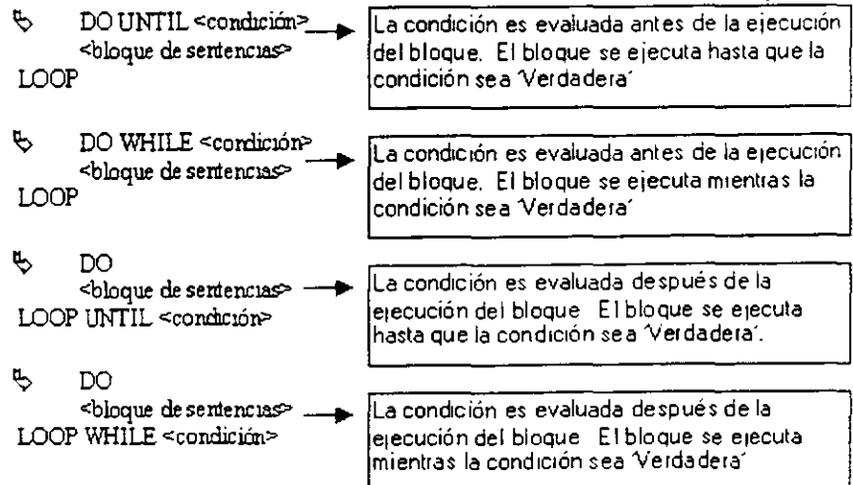
```
Next
```

En este ejemplo la sentencia se ejecutara cuando  $n \geq 5$  y  $n \leq 25$ , y el valor inicial de n (5) se incrementa en 5 por cada ejecución de las sentencias del bloque dentro de la sentencia for. El valor de Step puede ser negativo, en cuyo caso el valor inicial de la variable contador deberá ser inferior al valor final de la iteración.

### **DO..UNTIL | WHILE | LOOP**

La sentencia 'Do' es utilizada para repetir una(s) sentencia(s) un número no muy definido de veces. El número de veces es casi siempre definido por la comparación de una variable o expresión condicional.

Dentro de esta sentencia existen variantes, cuya única diferencia es el momento cuando se evalúa la expresión que condiciona la ejecución del bloque.



Ej.

integer A = 1, B = 1 //Emite un pito por los parlantes hasta que la variable

DO UNTIL A > 15 // A sea mayor que 15.

Beep(A)

A = (A + 1) \* B

LOOP

integer A = 1, B = 1 //Emite un pito por los parlantes mientras la variable

DO WHILE A <= 15 // A sea menor o igual a 15.

Beep(A)

A = (A + 1) \* B

LOOP

### Observación

En ambos tipos de iteraciones se puede terminar la ejecución del bloque utilizando la sentencia 'Exit'.

Ej.

```
FOR contador = 1 to 10
```

```
IF vector1[Contador] = 0 THEN EXIT
```

```
NEXT
```

En ambos tipos de iteraciones se puede continuar la ejecución del bloque, desde el principio y no ejecutando las demás instrucciones del bloque.

Ej.

```
integer A=1, B=1
```

```
DO WHILE A < 100
```

```
A = A+1
```

```
B = B+1
```

```
IF B > 3 THEN CONTINUE
```

```
MessageBox("Hola", "B es = " + String(B) )
```

```
LOOP
```

### **Utilización de Herramientas**

PowerBuilder posee un conjunto de herramientas destinadas a facilitar la creación de código fuente para un programa. Todas estas herramientas pueden ser utilizadas en cualquier módulo, donde se pueda escribir sentencias ejecutables.

Las Herramientas son:

- Listado de Funciones
- Sentencias SQL-Incrustado
- Listado de Sentencias(If, For, Do, etc.)
- Compilación
- Buscador

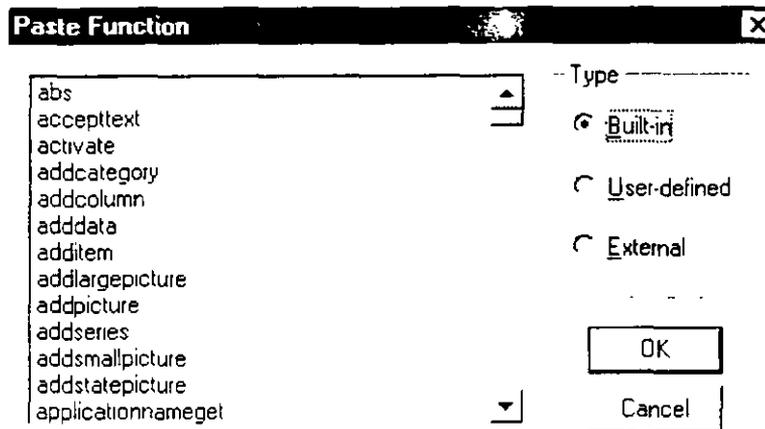
Además existen palabras reservadas destinadas a hacer referencia a objetos pertinentes, estas son:

	Descripción
<b>This</b>	Indica el objeto en el cual se esta codificando. Ej. this. Text = "hola"
<b>Parent</b>	Indica el objeto que contiene uno particular donde se esta codificando Close( parent )

## Listado de Funciones

Para accionar el listado de funciones presione el botón , a continuación aparecerá una ventana donde podrá seleccionar la función específica. Existen 3 agrupaciones (Funciones de PowerBuilder, Funciones de la Aplicación creadas por el Usuario y Funciones externas de DLL's conectados a la Aplicación).

Las funciones están ordenadas en forma ascendente, si desea buscar una función, escriba las primeras letras de esta, estando el foco en el listado de funciones.



Cuando seleccione la función, presione 'Ok' y esta se pegara en la ventana de edición del 'Script' en la ubicación del cursor actualmente.

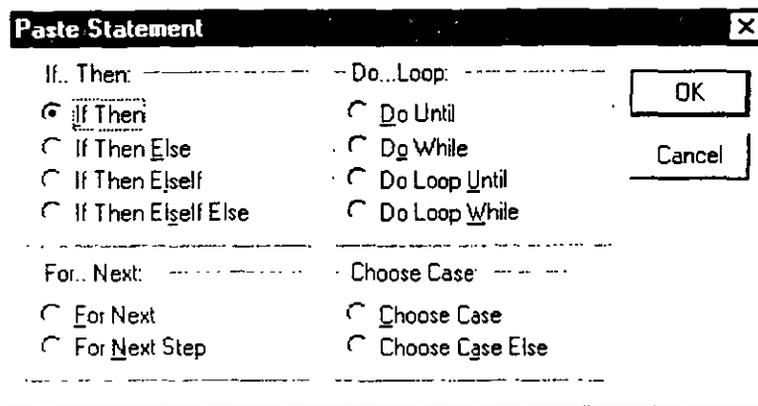
## Sentencias SQL-Incrustado

Para accionar las Sentencias SQL-Incrustado presione el botón , a continuación aparecerá una ventana donde podrá seleccionar el tipo de Sentencia SQL y cual específicamente( haciendo doble clic sobre la misma), o seleccionándola y presionando el botón 'Ok'.

Esta sentencia se pegara en el Script del modulo en el lugar donde este el cursor.

## Listado de Sentencias(If, Fôr, Do, etc.)

Para accionar el Listado de sentencias presione el botón, aparecerá una ventana donde podrá seleccionar la sentencia específica, solo una a la vez.



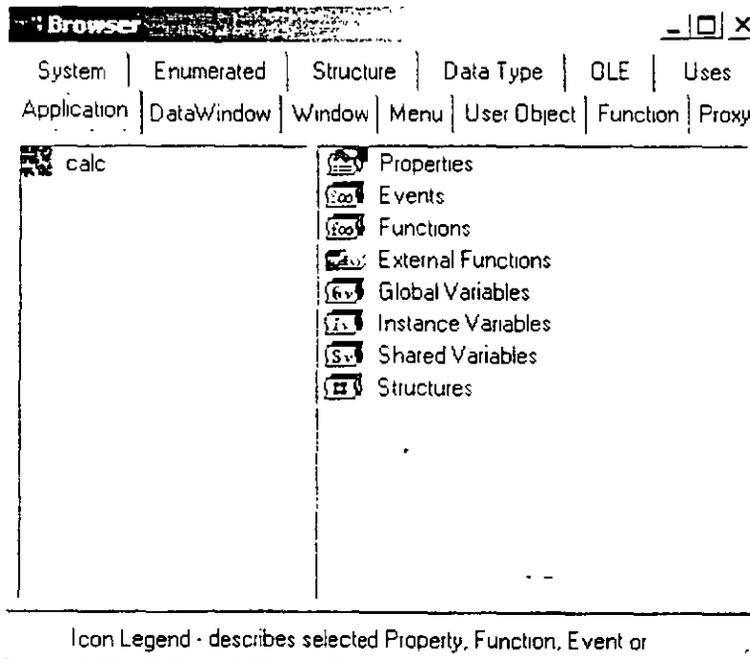
La sentencia se pegara en el Script del modulo, en el lugar donde este el cursor.

## Compilación

La compilación es el proceso durante el cual el modulo es revisado para encontrar errores sintaxis o léxicos. Para accionar este proceso presione el botón , en la parte inferior de la ventana del Script aparecerá un listado de los errores encontrados, si los hubiera.

## Buscador

El buscador permite obtener la información de los objetos que conforman la aplicación, información como; las funciones del usuario, las ventanas, datawindows existentes y todo objeto (atributos, funciones, etc.) que este presente en las ventanas.



Para accionarlo presione el botón , aparecerá una ventana como la anterior que tiene todos los tópicos por los cuales buscar un objeto.

Están organizados en forma jerarquizada y con subtópicos. A la mano izquierda están los objetos del grupo especificado, y a la derecha los tópicos por los cuales buscar. Seleccione un objeto, el tópico y el elemento específico; luego presione el botón 'Paste' y la sentencia especificada se pegara en donde este el cursor dentro de la ventana del Script

## Otras Herramientas

### Comentarios

Los comentarios, si es una línea se antepone a la mano izquierda dos 'slash' (//); eso significa que todo lo que este a la mano izquierda de estos dos signos será considerado como comentario.

Para poner comentarios; seleccione las líneas y presione el botón 

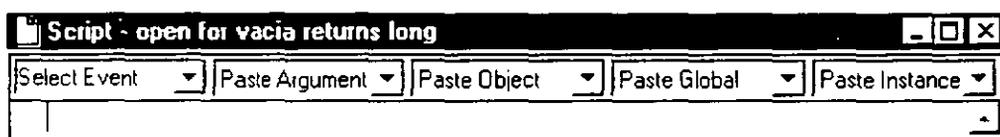
Para quitar los comentarios; seleccione las líneas y presione el botón 

### Herramientas de Edición

Podrá hacer uso de las opciones de edición utilizando los botones "Copiar" (, "Cortar" (, "Pegar" (); las que están presentes también en el menú 'Edit', o bien utilizando las teclas de acción directas.

### Listados en Módulos

Todo modulo ejecutable tiene una barra, más o menos, parecida a la siguiente.



En la Barra del titulo de la ventana aparece "Script", que quiere decir "Código". Al lado esta el nombre del modulo "Vacía", si fuese un objeto esta precedido por el nombre del Evento "Open" y luego el tipo de dato que puede ser devuelto por el modulo ejecutable.

Debajo de la Barra de Título, aparecen un conjunto de listas desplegables, si se selecciona un elemento que este contenida en una de ellas; este elemento (nombre) se pega en el código, donde este el cursor.

Si fuese un objeto, donde se esta escribiendo código, la primera lista es para seleccionar el evento donde se escribirá.

El listado "Paste Argument", permite pegar el nombre de un argumento del modulo en cuestión. El listado "Paste Object", permite pegar el nombre de los demás objetos contenidos en la ventana o menú, donde este el modulo en cuestión. El listado "Paste Global", muestra todas las variables/constantes declaradas como Global, y permite pegar cualquiera de ellas. El listado "Paste Instance", muestra todas las variables/constantes declaradas como Instance, y permite pegar cualquiera de ellas.

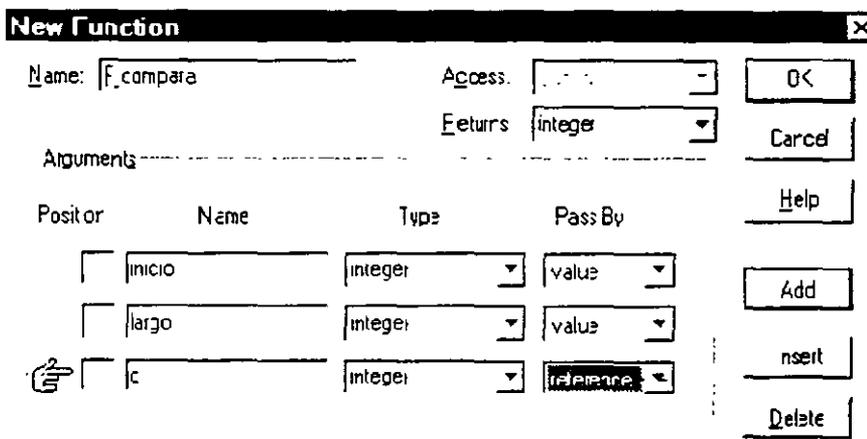
## Funciones

Las funciones en PowerBuilder pueden o no devolver un valor, este valor se conoce como "Valor de Retorno" de la función. Las funciones se declaran como elementos independientes y reconocibles en una librería, y puede ser utilizados desde cualquier parte de la aplicación.

Para trabajar con las funciones seleccione el icono  de la barra de herramientas 'PowerBar'. Aparecerá la ventana de selección de función; seleccione la función ya creada para seguir trabajando en ella, y luego presione el botón 'OK', o bien presione el botón 'New' si desea crear una nueva función.

## Declaración de Funciones

Al aparecer la ventana de selección de función, presione el botón 'New', aparecerá una ventana como la que sigue:



Positor	Name	Type	Pass By
<input type="checkbox"/>	inicio	integer	value
<input type="checkbox"/>	largo	integer	value
<input checked="" type="checkbox"/>	c	integer	reference

- 'Name', indica el nombre de la nueva función.
- 'Returns', indica el 'Valor de retorno', si la función no devolviera un valor de retorno, seleccione 'none'.
- 'Arguments', indica los argumentos (parámetros) de la función

## Parámetros

Una función puede o no tener parámetros, un parámetro es un valor o variable que es 'pasado' a la función para que la utilice en ella.

Al definir una función se declaran los parámetros que podrá tener. El mismo número de parámetros y el mismo tipo de dato para cada uno de ellos debe ser puesto al utilizar la función.

Ej `integer resp, 1`  
`resp = f_compara(1,2,i)`

Su paso fue definida por referencia, por ello debe ser obligatoriamente una variable

## Declarar los Parámetros

- Escriba el nombre del parámetro (nombre por el cual se hará referencia en el interior de la función) en la columna 'Name'.
- Seleccione o escriba el nombre del tipo de dato, en la casilla de la columna 'Type'.
- Seleccione el tipo de paso, en la casilla de la columna 'Pass By'.

Notara que cada parámetro tiene una posición definida en la columna 'Positor', esta misma posición debe estar al hacer la llamada a la función.

## Paso de Parámetros

Al declarar cada parámetro se debe indica como se 'pasara' el parámetro a la función, existen 3 tipos que son:

Tipo	Descripción
Value	El valor del parámetro pasado se copia al parámetro declarado, es decir, todo cambio que se haga al parámetro dentro de la función no alterara el valor del parámetro pasado.
Reference	El parámetro declarado esta relacionado con el parámetro pasado, es decir, todo cambio que se haga al parámetro dentro de la función, al terminar esta, quedara registrado en el parámetro pasado.
ReadOnly	Indica que el parámetro es de solo lectura (constante), no se podrá utilizar dentro del función como una variable.

Si la función requiere más parámetros presione el botón 'Add', este nuevo parámetro se anexara al final de los ya existentes. Si desea colocar un nuevo parámetro entre otros, presione el botón 'Insert'. Para eliminar un parámetro haga un clic sobre el y presione el botón 'Delete'.

### Valor de retorno

Elija el tipo de dato que será retornado al utilizar la función. Los tipos de datos pueden ser los definidos en el listado o bien escriba un tipo de dato que haya definido en la aplicación. Si el valor de retorno fuese asignado a una variable al llamarse a la función, tenga presente que dicha variable debe ser del mismo tipo del seleccionado en el campo 'Return'.

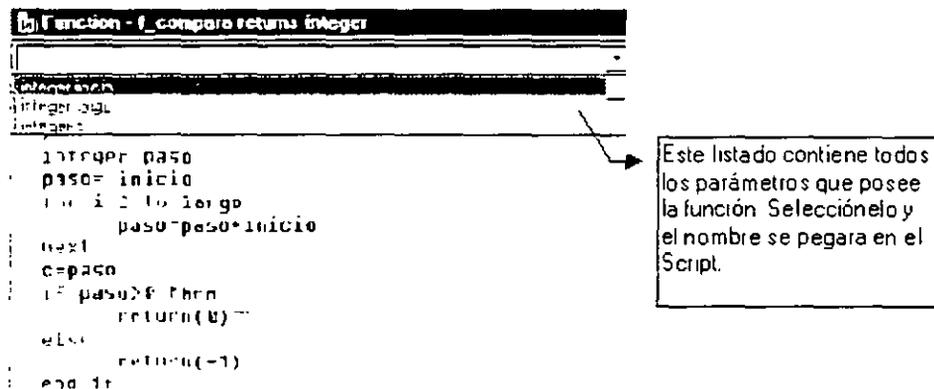
### Código Función

Dentro del código los parámetros son meras variables, y podrán ser utilizadas como tales (si no existen restricciones en su definición).

Se termina la ejecución de la función con la declaración 'Return'. Si se devolviera un valor esta deberá ir después de 'return' entre parentesis.

Se podrán utilizar las funciones primitivas, declaración de variables locales, flujos de control, etc.

Si desea modificar algo de la declaración de la función presione el botón .



The screenshot shows a window titled "Función - f\_compara retorna Integer". Below the title bar, there is a list of parameters: "Integer", "Integer", and "Integer". Below this list is a code block with the following content:

```
Integer paso
paso= inicio
for i = 1 to largo
    paso=paso+inicio
next
c=paso
if paso>f then
    return(u)
else
    return(-1)
end if
```

A callout box on the right side of the screenshot contains the text: "Este listado contiene todos los parámetros que posee la función. Selecciónelo y el nombre se pegara en el Script."

### Estructuras

Las estructuras de datos es una manera de organizar la información que esta contenida en un elemento para que sea más fácil su administración. Estas se pueden comparar con registros de datos donde, cada elemento particular de este es un campo. Los campos son elementos diferenciables uno de otro; como si fueran simples variables, pero; estando unidos por la definición de la estructura.

Ej.

Estructura: Persona

Campos	Tipo
nombre	String
edad	integer
sexo	char

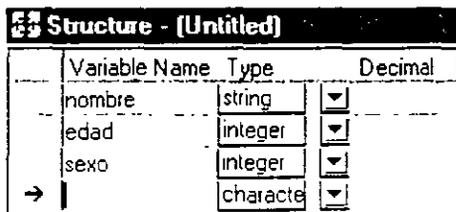
En las estructuras de datos no es posible definir reglas de validación tácitas, o restricciones de entradas. Como por ejemplo "Sexo='M' o Sexo='F'", estas deben ser definidas en el código fuente.

Los tipos de datos para cada campo pueden ser primitivos de Powerbuilder, objetos de Power o bien objetos definidos por el usuario.

Para trabajar con las estructuras presione el botón , al hacerlo; aparecerá la conocida ventana de selección. podrá elegir una estructura ya definida previamente, para continuar trabajando con ella, o bien podrá crear una nueva estructura presionando el botón 'New'.

### Definir Estructura

Después de presionar el botón 'New' aparecerá la ventana de 'definición de Estructuras'.



Variable Name	Type	Decimal
nombre	string	
edad	integer	
sexo	integer	
→	caracte	

Defina los campos que conformaran la estructura. En este caso el orden no es primordial.

- Escriba el nombre del campo, en la columna 'Variable Name', el que deberá ser único en toda la estructura.
- Seleccione o escriba el tipo de dato, en la columna 'Type'.
- Al salir de esta ventana se le pedirá que escriba el nombre de esta estructura, si es nueva; este nombre debe ser único para toda la aplicación.

Si necesita añadir un campo entre otros ya existentes, presione el botón  y continúe con la definición del campo. Si requiere borrar un campo, presione el botón ; teniendo cuidado que el cursor con la fecha → este ubicado en la casilla que corresponda al campo que desea borrar.

## Trabajar con Estructuras

Las estructuras que uno haya definido en el Área de trabajo de las Estructuras, pueden ser utilizadas en toda la aplicación; como si se tratara de un tipo de datos. Para poder hacer uso de la estructura, es imprescindible que primero se declare una variable, y el tipo de esta variable sea la estructura definida. También podrá utilizarse como tipo de dato de parámetro de una función, o bien de otra estructura.

Ej.

```
persona yo
```

```
yo.nombre= "Ricardo"
```

```
yo.apellido="Pavez"
```

```
yo.edad= 27
```

```
if yo.edad<= 20 then yo.descripcion= "Joven"
```

```
elseif yo.edad<= 65 then yo.descripcion= "Adulto"
```

```
else
```

```
yo.descripcion="Anciano"
```

```
end if
```

Como puede verse la variable 'yo' fue definida del tipo 'persona', que es una estructura diseñada por el Usuario. Para hacer referencia a cualquier campo se escritura <Variable>.<Campo>, el que puede ser utilizado para asignarle un valor a dicho campo (yo.edad= 27) o bien examinar el valor contenido en dicho campo (if yo.edad<=20 then ...)

Se vuelve a aclarar que dentro de la definición del campo de la estructura no es posible hacer una restricción de entradas, esta debe ser hecha en forma de líneas de código

## Menús

Los menús, para PowerBuilder son objetos. Las ventanas tienen una propiedad que indica cual objeto menú de los definidos para la aplicación hará uso, y ese será el menú que se anexará a la ventana. Hay que aclarar que lo que aquí se indica como 'menú', es en si una "Barra de Menús", que puede tener 1 o más ítems, y cada uno de estos ítems tiene un listado desplegable de opciones.

Para trabajar en el diseño de un menú presione el botón , que se ubica en la barra de Areas de Trabajo, y después seleccione el nombre del objeto menú con el desea continuar trabajando, o presione el botón 'New' para diseñar uno nuevo.

### Declaración, Uso

Después de presionar el botón 'New', aparecerá la siguiente ventana.

La ventana esta dividida en 3 secciones:

- **Items de la Barra de Menú**, la caja que aparece muestra el texto que será presentado como ítem de la Barra de menú. La barra de desplazamiento indica la posición dentro de la Barra de Menú, del ítem en particular.
- **Listado de Opciones**, las que aparecen bajo el titulo 'Menú For', cada caja es el texto que conforma una opción particular, para el ítem mostrado en la sección anterior.
- **Propiedades de la Opción**, son los tópicos mostrados a la mano derecha de la ventana, y corresponden a las propiedades de la opción que esta marcada del listado anterior con el símbolo de una mano.

Los tópicos que conforman la tercera sección son los siguientes:

Tópicos	Declaración
<b>General</b>	La casilla 'Menú Item Name', indica el nombre( lógico) que tendrá la opción en cuestión.La casilla 'MDI Microhelp' indica el texto que será mostrado como ayuda de la opción en una ventana declarada como tipo MDI.
<b>Style</b>	Indica las propiedades 'Visible', 'Enable'.La opción 'Checked', indica que si esta opción del menú es seleccionada aparezca al lado izquierdo de nombre una marca.
<b>ShortCut</b>	Indica la combinación de teclas para un acceso directo a la opción.Elija del listado 'Key' la tecla en particular y marque la tecla adicional (Alt, Ctrl, Shift)
<b>ToolBar</b>	Cada objeto Menú, puede tener asociado una caja de herramientas con botones para cada opción.La caja 'Text', indica el texto a ser mostrado junto con el botón definido para esta opción.La opción 'Visible', indica que el botón para esta opción sea mostrado en la caja de herramientas.La opción 'Display Down', indica que dicho botón aparezca con la r hundida.
<b>Pictures</b>	Determina las imágenes que serán utilizadas, cuando el botón este Normal, y cuando este Hundido. El botón 'Browse', permite seleccionar otra n de las existentes en el listado 'Stock Picture'. Estas ultimas tienen la cualidad de tener presentes las dos imágenes en una.

Para borrar la opción, o todo el Menú Item, presione el botón . Para Insertar una opción, o bien un Item de Menú, entre otros dos; presione el botón .

Para trabajar con las subopciones de una opción del Menú Item, presione el botón , y para regresar al nivel anterior presione el botón .

### Programación de Acción

Para trabajar codificando acciones presione de la barra de herramientas secundaria el botón , el objeto menú, para cada opción permite definir código en dos Eventos, los que son:

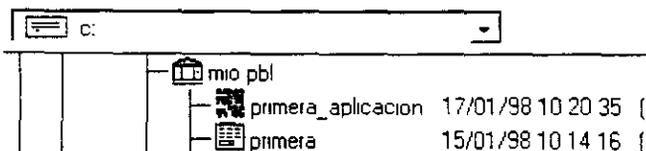
Evento	Descripción
<b>Cliced</b>	Sobre esta opción el Usuario ha presionado el botón del Ratón, o presiono Enter, o tambien se ha presionado el botón que fue definido para esta opción.
<b>Selected</b>	Cuando una opción es seleccionada por el ratón (se pasa por encima de ella).

Puede hacerse uso de todas las características de PowerScript.

### Librerías

Permite distinguir los elementos que conforman una librería. Estas no están distribuidas en forma jerárquica como en el área de "Aplication", solo son presentadas en orden del Area que forman parte.

Para trabajar en el área de Librerías presione el botón . Aparecerá una ventana como la siguiente



Donde en la parte superior se selecciona la unidad a revisar existencia de Librerías, y en la parte inferior una especie de directorio. Los archivos de librería están acompañados por el icono .

- **Crear Librería**, para ello presione el botón , escriba el nombre e indique la ruta de la nueva librería.

Las siguientes operaciones, se pueden realizar con uno o más elementos seleccionados, de una Librería:

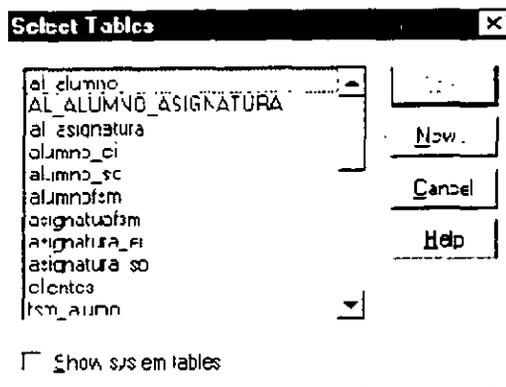
- **Editar**, permite iniciar el área a la que pertenece el elemento; presione el botón .
- **Copiar**, presione el botón  y elija la librería donde copiar los elementos seleccionados.
- **Mover**, presione el botón  y elija la librería a donde moverá los elementos seleccionados.
- **Eliminar**, presione el botón  para eliminar el (los) elementos marcados.
- **Exportar**, permite extraer la definición del elemento como un archivo texto; para ello presione el botón .
- **Importar**, permite añadir la definición de un elemento desde un archivo texto a la librería en curso; para ello presione el botón .
- **Regenerar**, permite que se actualicen los vínculos con otros elementos, modificados, de PowerBuilder o de una Base de Datos, para ello presione el botón .
- **Propiedades**, Permite modificar el comentario declarado para cada elemento; para ello presione el botón .

Todos estas opciones están disponibles en el menú 'Entry'

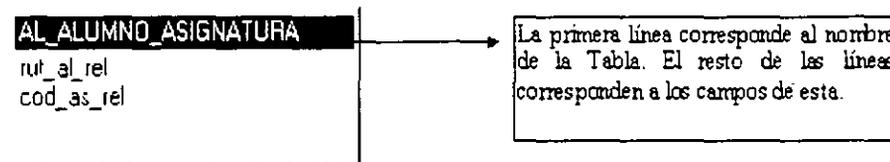
## Base de Datos

El área de Base de datos permite crear nuevas Bases de datos SQL- Anywhere, crear/modificar/eliminar Tablas, actualizar campos de dichas Tablas. También permite definir Estilos de presentación/Reglas de validación/Formatos de Datos; todos estos para ser utilizados por cualquier campo de las Tablas de la Base de Datos.

Para trabajar en el Area de Base de Datos, presione el botón , se conectara a la Base de Datos definida por defecto, en el Area Profile Database. Si esta Base de datos no fuese la suya seleccione la suya en el Area Profile Database, presionando el botón  y vuelva a presionar el botón de Base de datos. Después de conectarse correctamente a al Base de datos aparecerá la ventana de "Selección de Tabla", solo las tablas que se seleccionen del listado será posible de manipular.

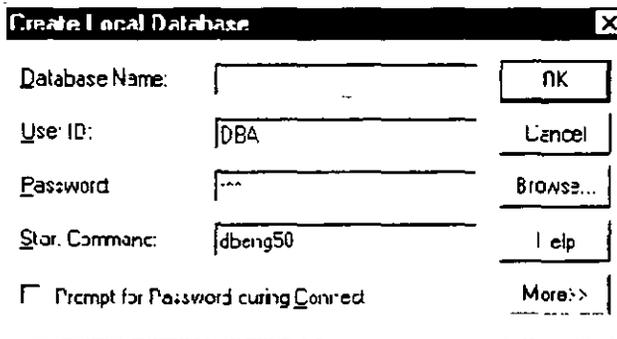


Para marcar una tabla haga un clic sobre su nombre, para desmarcala vuelva a presionar el ratón sobre ella. Finalmente presione el botón 'Open' para que aparezcan las tablas seleccionadas, o presione el botón 'Cancel' para cancelar la operación. La opción 'System Tables' permite seleccionar las Tablas del catalogo de la Base de Datos; estas Tablas son utilizadas para la Administración de la Base de Datos. Las Tablas seleccionadas aparecen en la ventana en una forma gráfica



### Nueva Base de datos

Para crear una base de datos local (SQL- Anywhere) nueva, estando en el área de Base de datos seleccione la opción 'New Database' del menú 'File'. Aparecerá una ventana



En la casilla 'Database Name', escriba la ruta y el nombre de la nueva base de datos, o presione el botón 'Browse' para hacer esta operación. En la casilla 'User ID' escriba el nombre del usuario, por defecto se asigna "DBA". En la casilla 'Password' escriba la contraseña correctamente, esta no es visible, por defecto se asigna "sql".

Cuando haya terminado presione el botón 'Ok', espere a que se realicen todas las operaciones de creación de la Base de datos. Aparecerá un mensaje indicándole el resultado de la creación. Si esta fuese satisfactoria aparecerá la ventana de "Selección de Tabla". Presione el botón 'New ' para crear una nueva Tabla.

## Tablas

La tabla de una base de datos es una colección de información homogénea (relacionada entre si). Los elementos de esta información se denominan registros, cada registro esta conformado por idéntico número y tipo de campos. Cada campo tiene asociado un nombre; por el que se identifica, y un tipo de dato, que especifica la información que podrá almacenar dicho campo.

Una Tabla puede tener 0 o más registros y la definición de un registro deberá contar con a lo menos un campo.

Ej. registro *persona*

Rut	Nombre	Apellido	Domicilio	Sexo	Fech. Nac	Naciona
125537987	raul	yañez	cruz 2356	masculino	20/12/1970	chilena
:	:	:	:	:	:	:
96789870	marta	gutierrez	heras 45	femenimo	6/9/1968	argentina

## Nueva Tabla

Para crear una nueva Tabla presione el botón . Aparecerá una ventana que tiene 2 secciones, esta ventana se llama "Ventana de Definición de Tabla".

Esta ventana esta dividida en dos secciones, superior e inferior.

### a) Definición de Campos

En esta sección se declaran los campos, una línea para cada uno de ellos.

### b) Atributos adicionales (Extended Attribute)

En esta sección se añade información, como es, el estilo de presentación, regla de validación, dimensiones del área de ingreso de datos para cada campo, etc., sobre el campo actual, el campo actual es aquel donde este el símbolo de la mano apuntando en la sección de "Definición de Campos".

Si presiona el botón  se procederá a grabar la definición de la Tabla, se le preguntara el nombre que se le dará, este nombre debe ser distinto al de las otras tablas de la Base de datos.

Si cerrara la ventana de definición de Tabla, sin guardar las modificaciones hechas, se le preguntara si desea guardar los cambios. Si no se le asigno un nombre a la Tabla previamente, deberá ingresarlo. Después de esto regresara a la ventana de representación gráfica de las Tablas.

Presione el botón  y podrá escribir el nombre de la Tabla, y definir en los diversos tópicos, la clave primaria, claves foráneas, e índices. (todo esto se explicara a continuación)

Vamos a explicar cada uno de los elementos que conforman cada sección en el mismo orden en que fueron nombradas:

### Definición de Campos

Esta sección esta dividida con la siguiente estructura:

Columna	Descripción
<b>Name</b>	Define el nombre del campo.
<b>Type</b>	Seleccione de la lista desplegable el tipo de dato que será almacenado en el campo.
<b>Width</b>	Si selecciono en la columna anterior el tipo de dato Char, Varchar o Numeric. Deberá especificar el largo (número de caracteres/dígitos) máximo a ser almacenados en el mismo.
<b>Dec</b>	Si selecciono en la columna 'Type', el tipo de dato Numeric. Podrá indicar el número de decimales que serán utilizados para este campo. En el número que se escribió en la columna 'Width' están contenidos el número de decimales aquí especificados.
<b>Null</b>	Indica si se permitirá que este campo tenga un valor NULL.
<b>Default</b>	podrá seleccionar (de la lista desplegable) un valor por defecto para este campo, o bien, escribirlo como un literal.

Si desea insertar una columna entre otras ya existentes presione el botón , si necesita borrar una columna presione el botón .

### Atributos adicionales

Esta sección permite definir atributos que tendrá el dato del campo al ser mostrado al usuario. La información corresponde al campo actual, cada campo podrá tener diversos valores para estos atributos. Los atributos de esta sección son los siguientes:

<b>Atributo</b>	<b>Descripción</b>
<b>Format</b>	Seleccione de la lista desplegable el formato diseñado previamente. Este formato es una regla para la aceptación de un campo a partir de una revisión de los caracteres que lo conforman.
<b>Edit</b>	Seleccione de la lista desplegable el Estilo de Edición diseñado previamente. Este estilo permite restringir los valores a tomar por un campo (lista desplegable, casillas de selección, etc.) o una máscara de entrada de datos.
<b>Valid</b>	Seleccione de la lista desplegable la Regla de Validación diseñado previamente. Esta regla de validación permite comprobar si el dato ingresado cumple con una condición fijada.
<b>Header</b>	Será el texto que aparecerá como identificación del campo al ser editable en forma de columnas.
<b>Label</b>	Será el texto que aparecerá adjunto a un campo en un Datawindows.
<b>Justify</b>	Indica el tipo de justificación que tendrá el dato del campo. Seleccione uno de la lista desplegable.
<b>Height</b>	Indica la altura y anchura del campo. Estos valores serán utilizados para definir el ancho y el alto del espacio destinado al ingreso/selección de datos para este campo.
<b>Width</b>	
<b>Initial</b>	Es el valor que será definido como inicial para un nuevo registro de la Tabla.
<b>Comment</b>	Permite escribir un comentario adicional sobre el campo en particular. Este texto aparecerá en la representación gráfica de la Tabla.

### Propiedades Tabla

La ventana Propiedades de la Tabla, contiene los siguientes tópicos, algunos no serán mostrados sino esta en la ventana "Definición de Tabla"; todos ellos muy importantes:

<b>Topico</b>	<b>Descripción</b>
<b>General</b>	Especifica el nombre de la Tabla, y algún comentario adicional.
<b>Data Font</b>	Permite definir las características de tipo de letra, y otras; que serán utilizadas para presentar el dato específico del campo al usuario, y los Atributos adicionales 'Label' y 'Header', respectivamente.
<b>Heading Font</b>	
<b>Label Font</b>	
<b>Heading Font</b>	Permite definir las características de tipo de letra, y otras; se serán utilizadas . -
<b>Indexes</b>	Un listado de los Indices definidos para esta Tabla
<b>Primary Key</b>	La clave primaria definida para esta Tabla.
<b>Foreign Key</b>	Un listado de las claves foráneas definidos para esta Tabla.

Para grabar los cambios hechos en los tópicos de la ventana Propiedad de la Tabla, presione el botón 'Apply' para no abandonar la ventana de propiedades y el botón 'Ok' para abandonarla y regresar a la ventana de definición de Tabla. Presione el botón 'Cancel' para deshacer las modificaciones hechas en esta ventana.

### **Modificar Tabla**

Estando en el área de Base de datos, haga doble clic sobre la representación gráfica de la Tabla a modificar.

Si esta no estuviera presente deberá abrirla agregándola a las ya presentes. Presione el  y aparecerá la ventana de selección de Tablas, haga su selección y presione el botón 'Open'.

Al hacer doble clic sobre la representación gráfica de la Tabla aparecerá la ventana de definición de Tabla, definida previamente. Estando aquí realice las modificaciones que quiera en la definición de los campos, (algunos aparecen inhabilitados; ya que no pueden ser modificados después de haber sido definidos) en los atributos adicionales, o bien en las propiedades de la Tabla.

Para grabar los cambios hechos presione el botón  o cierre la ventana y conteste 'Sí' a la pregunta de guardar los cambios.

### **Eliminar Tabla**

Para eliminar una Tabla, su representación gráfica debe estar presente, si no es así; añádala. Haga un clic sobre la Tabla a eliminar, y presione el botón .

Tenga presente que junto con eliminar la Tabla se eliminarán los registros que estén en ella, los índices definidos, la clave primaria y las claves foráneas.

### **Clave Primaria**

La clave primaria es el(los) campo(s) que permiten identificar a un registro. Para definirla debe estar en la ventana de Definición de Tabla ( Haga doble clic sobre la representación gráfica de la Tabla), y presione el botón , elija el tópico 'Primary Key'.

Key Columns:

Table Columns:                      Key Name

(Untitled)

Los campos marcados del listado 'Table columns' son los que forman la clave primaria.

Haga un clic sobre un campo no marcado y este pasara a formar parte de la clave primaria, haga un clic sobre un campo ya marcado y dejara de formar parte de la clave primaria. En la casilla 'Key columns' se puede observar gráficamente los campos que forman la clave primaria.

Digite el nombre de esta Clave Primaria, en la casilla 'Key Name', si fuese necesario definirle un nombre.

Al volver a la ventana de representación gráfica de las Tablas, aparecerá un símbolo gráfico  adjunto a la Tabla, esto indica que esta tabla tiene una Clave primaria, las líneas que nacen de la representación gráfica de la Tabla, indican el(los) campo(s) que forman parte de la clave primaria.

### Clave Foránea

La clave foránea es un(os) campo(s), cuyo valor(es) deben estar presente en otra Tabla y los campos de esta segunda Tabla deben ser (o formar parte) de la Clave primaria. Esto último permite definir una relación de integridad entre los datos; la existencia de el dato en la primera Tabla esta condicionado a la existencia de el(los) mismo(s) dato(s) en la segunda Tabla.

Para definir una clave foránea debe estar en el ventana de Definición de Tabla (haga doble clic sobre la representación gráfica de la Tabla)

Estando en la ventana de "Definición de Tabla", presione el botón  y elija el tópico 'Foreign Key'.

Foreign Keys.

fk_personal_persona	Edit..
	New..
	Delete

En el listado aparecen los nombres dados cada relación, ya que una Tabla puede tener 0 o muchas relaciones (claves foráneas).

Marque una relación del listado, presione 'Edit' para modificarla, 'Delete' para borrarla. Presione el botón 'New' para definir una nueva relación.

Foreign Key Name.	Primary Key Table:
<input type="text"/>	<input type="text"/>
Foreign Key Columns	Primary Key Columns.
<input type="text"/>	<input type="text"/>
<input type="button" value="Select Columns."/>	<input type="button" value="Select Columns."/>

Los detalles para desarrollar una Clave Foránea son:

- En la casilla 'Foreign Key Name', es el nombre dado a esta relación.
- El(los) campo(s) marcados del listado 'Select Columns' conformaran la Clave Foránea.
- La caja 'Primary Key Table' indica la Tabla con la que estará relacionada En la parte inferior de esta casilla aparecerá el campo(s) que conforman la clave primaria de esta Tabla, 'Primary Key Columns'. El número de campos seleccionados previamente deberá ser el mismo a los que aparecen en esta casilla. El primer campo seleccionado de la Tabla estará relacionado con el primero de la Tabla elegida, y todos los demás campos seleccionados de igual forma. No es necesario que tengan el mismo nombre los campos relacionados, pero si deben ser del mismo tipo de dato.

Al regresar a la Representación gráfica de las Tablas, aparecerá un símbolo gráfico  adjunto a la Tabla, esto indica que esta tabla tiene una Clave Foránea definida, las líneas que nacen de la representación gráfica de la Tabla, indican el(los) campo(s) que forman parte de la Clave Foránea y la Tabla con la que están relacionadas.

Haga doble clic sobre el símbolo adjunto y podrá ver información sobre la Clave Foránea definido.

## Índices

Los índices se definen para mejorar la búsqueda y selección por parte del motor de Base de Datos. Una Tabla puede tener varios índices definidos, o tener ninguno. Cada índice se crea para uno o más campos, estos últimos se llaman "Índices compuestos".

Para definir Índices, debe presionar el botón de Propiedades de la Tabla (estando en la representación gráfica de las tablas o en la ventana "Definición de Tabla"), elija el tópico 'Index'.

Aparecerá un listado con todos los índices definidos para esta Tabla.



Marque una relación del listado, presione 'Edit' para modificarla, 'Delete' para modificarla. Presione el botón 'New' para definir una nueva relación.



Los detalles para crear un Índice son:

- En la casilla 'Index Name', aparecerá el nombre que se le dará a este Índice.
- El(los) campo(s) marcados que aparecen en la parte inferior de la ventana que conformaran el Índice.
- La opción 'Unique'/'Duplique' indica si los campos que forman el índice son Unicos o están Duplicados dentro de los registros de la Tabla.

Al regresar a la Representación gráfica de las Tablas, aparecerá un símbolo gráfico, con forma de llave, adjunto a la Tabla, esto indica que esta tabla tiene un Índice definido, las líneas que nacen de la representación gráfica de la Tabla, indican el(los) campo(s) que forman parte del Índice.

## Actualizar Datos Tabla

Para trabajar con los datos de la Tabla, los modos de presentación de la información más comunes son:

- En forma de malla, los campos son las columnas y las filas los registros. Para trabajar en esta forma presione el botón .
- En forma libre, los campos son listados hacia abajo uno a continuación del otro y los registros de igual forma. Para trabajar en esta forma presione el botón .

Estando en el modo "Actualización Datos Tabla":

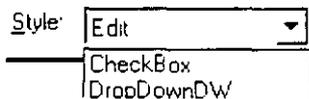
- Si se desea añadir un nuevo registro al final de los ya existentes presione el botón .
- Para eliminar un registro presione el botón , se eliminara el registro donde este el cursor actualmente.
- Para guardar los cambios en la Base de datos , presione el botón .
- Para obtener datos desde la Base de Datos, presione el botón .
- Para terminar el modo "Actualización ", presione el botón .
- Para moverse de un campo a otro presione la tecla 'Tab'. Para moverse de un registro a otro utilice las teclas cursor arriba() y cursor abajo() , para ir un registro antes y un registro después respectivamente.

## Estilos de Edición

Los estilos de edición (presentación de los datos) son elementos que pueden ser utilizados en varias Tablas a al vez. Para trabajar con los diseños de presentación, elija la opción 'Edit Style Maintenance', del menú 'Design'. Aparecerá una ventana con un listado de todos los Estilos definidos. Si desea continuar definiendo uno, presione 'Edit', para borrar uno definido presione 'Delete', y para crear uno nuevo presione 'New'.

## Diseño Estilo

Aparecerá una ventana para el diseño del estilo.'Name', indica el nombre de este estilo. La lista desplegable 'Style' muestra los tipos permitidos.



Style	Descripción
Edit	Muestra el dato en una caja de edición, esto permite que pueda ingresar un dato vía el teclado. La opción 'Display Only' indica que el campo es de solo lectura. La opción

	<p>'Empty String..' indica que la caja vacía será reconocida como Null. La opción 'Password' muestra los caracteres escritos con el símbolo visual '*'. Las opciones 'Auto..' permiten que aparezca una barra de desplazamiento si el número de caracteres supera a los visibles en el largo de la caja.</p>
<b>EditMask</b>	<p>Este estilo también muestra el dato en una caja de edición, esto permite que pueda ingresar un dato vía el teclado, pero además con una mascara de entrada; esto es, una restricción carácter a carácter de los grupos de letras posibles o la forma en que serán mostrados. En el casillero 'Mask' define la mascara de entrada. <b>Ej.</b> El rut: ##.###.###-! ( esto permite que se ingresen 8 caracteres numéricos y un alfanumérico en mayúscula, los otros símbolos se entienden por constantes). El listado 'Masks' muestra los símbolos que se utilizan para generar la mascara de entrada, puede probarla escribiendo un dato de prueba en la casilla 'Test'. La opción 'Type' indica el tipo de dato del campo. 'Autoskip' significa que cuando se llene el campo el cursor ira al siguiente campo. 'Spin Control' permite que aparezca dos botones, para aumentar el valor y disminuir el valor que este en el campo, este aumento será definido en la opción 'Spin increment', se debe definir en 'Min' el valor minino de incremento y en 'Max' el valor máximo de incremento.</p>
<b>DropDownListBox</b>	<p>Este estilo permite seleccionar/mostrar un valor dentro de una lista de ítems definida. En la caja 'Display value' escriba el valor a mostrar realmente y en 'Data value' el valor a ser almacenado en el campo cuando se elija esa opción en el listado. Ocupe los botones 'Add' para añadir un nuevo ítem, 'Insert' para insertarlo antes de otro y 'Delete' para borrar el actual ítem.</p>
<b>RadioButtons</b>	<p>Este estilo permite seleccionar/mostrar un valor de los listados como opciones separadas. 'Left Text' muestra el texto al lado izquierdo de la casilla de selección. '3D Look' muestra la casilla de selección grande. En la caja 'Display value' escriba el valor a mostrar realmente y en 'Data value' el valor a ser almacenado en el campo cuando se elija esa opción de entre las mostradas separadamente. Ocupe los botones 'Add' para añadir un nuevo ítem, 'Insert' para insertarlo antes de otro y 'Delete' para borrar el actual ítem.</p>
<b>DropDownDW</b>	<p>Este estilo de edición es particular. Se ocupa para aquellos campos de la Base de datos que deban tener el mismo valor que otro campo de otra Tabla; es decir, sean campos de <i>Clave Foranea</i>.</p>

	<p>Lo primero es crea un Datawindows que presente como mínimo la clave primaria de la Tabla a la que hace referencia la Clave Foránea. Este datawindows no debe tener información en la sección 'Header' de la presentación, tampoco deberá tener agrupaciones, ni objetos compute.</p> <p>Luego en el Datawindows que este diseñando y la casilla 'Style' elija la opción 'DropDownDW'. Las 3 cajas de texto, indican:</p> <p>En 'DataWindows', el datawindows que creo previamente. En la opción 'Display Column' el campo que servirá para ser mostrado solamente. 'Data Column' el campo que servirá para guardar el dato realmente.</p> <p>En a opción 'Lines in ..' se indica el alto de la caja desplegable.</p>
--	---

### Reglas de Validación

La reglas de validación son elementos que pueden ser utilizados en varias Tablas a al vez. Para trabajar con estas, elija la opción 'Validation Maintenance', del menú 'Design'. Aparecerá una ventana con un listado de todas las reglas definidos. Si desea continuar definiendo uno, presione 'Edit', Para borrar uno definido presione 'Delete', y para crear uno nuevo presione 'New'.

La regla de validación es revisada cuando el cursor esta en el campo y se mueve de este a otro; ya sea del mismo registro u otro.

### Diseño Regla

La casilla 'Name', indica el nombre de la regla de Validación. Defina el tipo de datos que manipulara la regla, 'Type'. El botón 'Match' le permitirá definir una sentencia de consulta sobre cada carácter ingresado.

Validation Expression:

```
msgid > 0 and len(gettext())=2
```

En la caja 'Validation Expression' deberá escribir la regla de validación.

Para hacer referencia, en la regla de validación del campo que se esta ingresando, presione el botón "'@Col"

En el listado 'Functions' aparecen todas las funciones predefinidas que puede utilizar, haga un clic sobre alguna y esta se pegara en la caja 'Validation ..'. La letra que aparece como parámetro de la función tiene un significado.

	<b>Descripción</b>
x	Significa que debe ser reemplazado por el nombre de un campo, o bien por un valor dependiendo del tipo de función.
#x	Indica que debe ser reemplazado obligatoriamente por el nombre del campo.
s	Indica que debe ser reemplazado por un valor texto o bien por el nombre de un campo de tipo texto. Si se utiliza un valor texto literal escríbalo entre comillas simples (') <b>Ej.</b> Len('hola')=4
n	Indica que debe ser reemplazado por un valor numérico o bien por el nombre del campo de tipo numérico.
d	Indica que debe ser reemplazado por un valor fecha o bien por el nombre del campo de tipo fecha.
T	Indica que debe ser reemplazado por un valor tiempo o bien por el nombre del campo de tipo tiempo (hora).

En la regla de validación se pueden utilizar operadores relacionales (AND, OR) para separar sub-expresiones. También se pueden utilizar todos los operadores matemáticos.

La función *gettext()* obtiene el valor que esta en la caja de edición como un dato texto, no importando el tipo de dato definido para dicho campo.

### **Objeto de Transacción**

Un objeto de transacción es un objeto de PowerBuilder, y cumple la función de intermediario entre la aplicación en si, y el administrador de Base de Datos.

Cuando usted hace una nueva aplicación, se genera un objeto de transacción por defecto. Su nombre es 'SQLCA'.

### **Nuevo objeto de transacción**

Podrá ser necesario utilizar otro objeto adicional al por defeco, para declarar un nuevo objeto de transacción escriba:

Transaction <nuevo obj>

Donde:

- <nuevo obj>, es el nombre de la nueva variable.

Ej.

**Transaction** sqlca2

### Trabajar con Objeto Transacción

- Todo objeto de Transacción, requiere parámetros para trabajar con el Administrador de Base de Datos.

Ej.

```
sqlca.dbms= 'ODBC'
```

```
sqlca.dbparm= "ConnectionString= 'dsn=  
example;uid= dba; pwd= sql'"
```

- Antes de utilizar el objeto de transacción, en cualquier sentencia o Datawindow, necesita conectarse a la Base de Datos.

Ej.

```
Connect using sqlca2;
```

- Después de realizar una sentencia, es recomendable comprobar si ha ocurrido algún error al ejecutar una sentencia SQL.

Ej.

```
If sqlca.sqlcode = -1 Then
```

```
    messagebox ( "Error" , "SQL erro: "+  
error.text )
```

```
End If
```

- Los cambios hechos a la Base de Datos deben traspasarse permanentemente a esta, por si ocurriera un error.

Ej.

```
Commit Using sqlca2 ;
```

- Si un error ocurre al ejecutar una sentencia SQL, es necesario deshacer las modificaciones hechas.

Ej.

**Rollback Using sqlca2 ;**

- Al terminar la aplicación, es necesario cerrar la sesión con la Base de Datos.

Ej.

**Disconnect Using sqlca ;**

## DataWindows

Un DataWindows es una sentencia SQL que esta ligada a una representación gráfica. Donde cada elemento de esta representación es individualizable y puede ser alterado en el código del programa.

Esta definición indica dos elementos unidos (aunque distintos entre si)

### a. **Origen de Datos:**

Corresponde a la sentencia SQL propiamente tal, la que podrá tener características de Agrupamiento, Ordenamiento, Condicionales (extraer un conjunto definido de registros) y en fin todas las pertinentes a una sentencia SQL pura.

### b. **Presentación:**

Corresponde a la distribución gráfica de los elementos de la consulta y otros añadidos, dentro de un área definida para ello. La que podrá ser utilizada para presentar datos, ingresar datos, listarlos o imprimirlos; tal y como aparecen en esta hoja.

## Selección Tipo y Presentación

Para trabajar con las Datawindows presione el botón , luego de lo cual aparecerá la ventana de selección; podrá seleccionar una ya existente, para modificarla, o presionar el botón 'New' para crear una nueva Datawindows.

Si presiona el botón 'New' deberá seleccionar el modo de trabajo para el 'Origen de datos' y también el tipo de 'Presentación'.

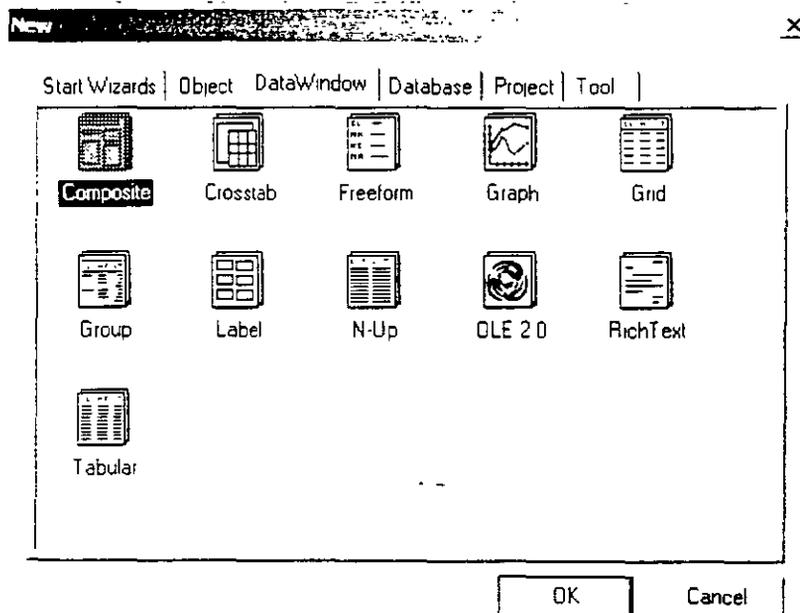
## Modos de trabajo:

El modo de trabajo define como se trabajara con el Origen de Datos. Existen 5 tipos de Modos de trabajo.

Modo	Descripción
<b>Quick Select</b>	No presenta las tablas en forma gráfica, y solo permite las sentencias ordinarias de SQL (Sort y Criteria) La primera para poder ordenar por el campo específico y la segunda para indicar registros a mostrar dependiendo de un valor condicional ( <b>Ej.</b> <39)
<b>Sql Select</b>	Presenta las tablas en forma gráfica, y permite trabajar con todas las características de Sql estándar (Sort, Group, Having, Compute y Where, entre otros)
<b>Query</b>	El origen de Datos es una Consulta definida previamente en el Área de Trabajo Query; esta consulta queda almacenada como un objeto de Powerbuilder; debe seleccionar la consulta que desea utilizar como Origen de Datos de este Datawindow.
<b>External</b>	El origen de datos no está relacionado con ninguna Tabla. debe definir los campos que interactúan (nombre, tipo y largo) Utilice los botones 'Add' para añadir un nuevo campo al final de los ya existentes, el botón 'Insert' le permitirá insertar un campo entre otros y el botón 'Delete' le permitirá eliminar el campo actual.
<b>Stored Procedure</b>	El origen de datos es un procedimiento que ha sido almacenado en la Base de Datos. Seleccione uno si lo ha creado, o créelo previamente.

## Tipos de presentación

El Tipo de presentación define como se presentaran los datos en el DataWindows, existen 11 tipos de presentación las que son:



<b>Presentación</b>	<b>Descripción</b>
<b>Composite</b>	Utiliza como presentación un reporte previamente creado.
<b>Freeform</b>	La presentación es libre, donde cada campo esta en una línea distinta. Muy utilizado para el ingreso de datos.
<b>Graph</b>	Diseña un gráfico dependiendo de una consulta, esta debe contener datos relacionados.
<b>Grid</b>	Diseña una malla de datos en forma de columnas, y cada columna con una justificación completa.
<b>Group</b>	Diseña listado por agrupación de campos. Muy utilizado para reportes de resumen de información.
<b>Label</b>	El diseño es de etiquetas, donde podrá elegir el tipo de etiqueta de las prediseñadas, en papel continuo o separado, y otras opciones.
<b>Rich Text</b>	Permite definir una carta con combinación de campos del Origen de datos fijado. Se pueden utilizar todas las características del formato de texto RTF.
<b>Tabular</b>	Presenta el origen de datos en forma de columnas pero sin ajuste total. Permite una mayor libertad en el diseño.
<b>Crosstab</b>	Un crosstab permite a los usuarios ver datos sumarios en lugar de una serie larga de filas y columnas. Por ejemplo, en una aplicación de las ventas usted podría querer resumir el trimestral las ventas de la unidad de cada producto.
<b>N-Up</b>	Es similar al estilo de Label en el cual puede tener información de varias filas en el banco de datos por página. Sin embargo, la información no significa que se imprima en etiquetas. El estilo de esta presentación es útil si tiene datos periódicos.
<b>OLE 2.0</b>	Le permite vincular un objeto OLE en un objeto de DataWindow.

## Origen de Datos

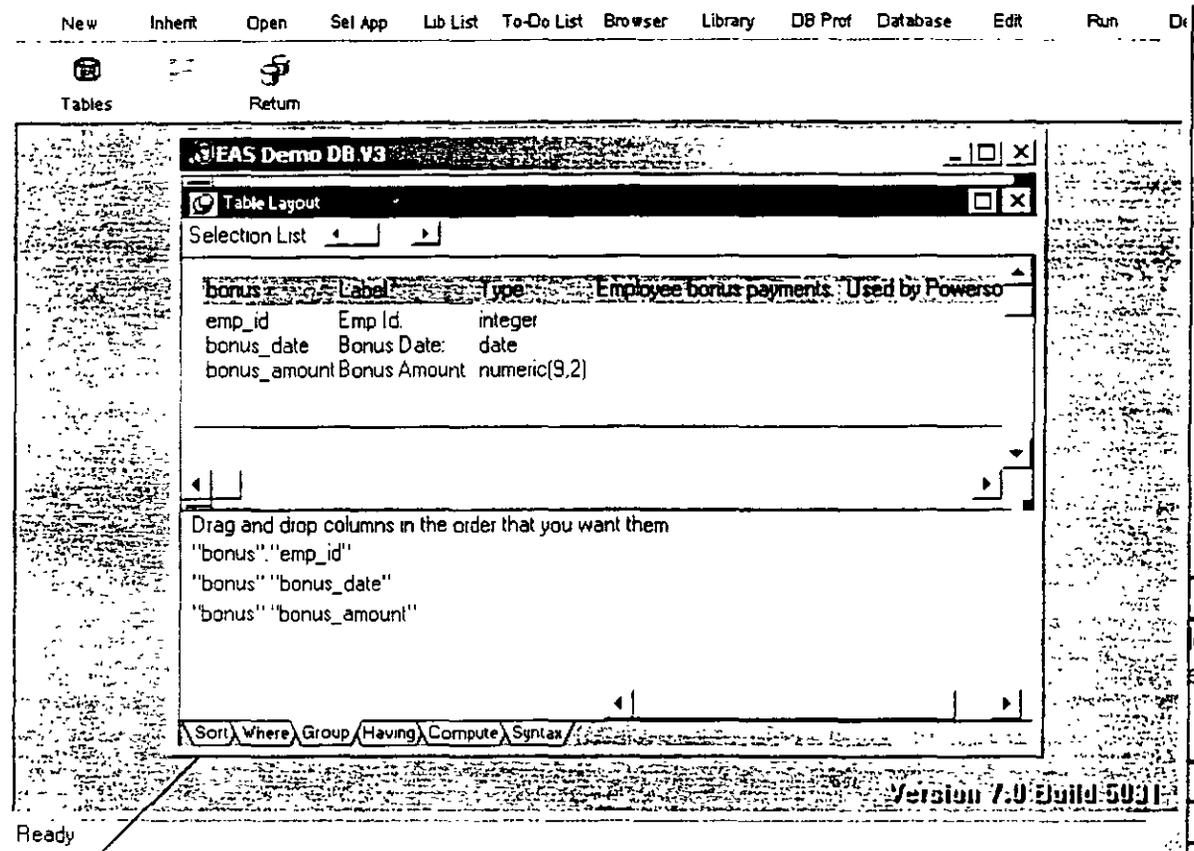
Después de seleccionar una de las opciones de DataWindows, aparecerá un conjunto de opciones que permitirán seleccionar la(s) Tabla(s) que estarán involucradas si este es nuevo.

Después de elegir los campos requeridos, y en algunos casos el formato de la tabla, se procede a finalizar el asistente; si desea cancelar la selección presione el botón 'Cancel', luego de lo cual aparecerá en forma gráfica la(s) Tabla(s) y también las relaciones existentes entre ellas. Si deseara añadir nuevas tablas presione el botón  y selecciónela(s). Si deseara extraer una tabla del Origen de datos, presione el botón derecho del ratón sobre la representación gráfica de la Tabla y seleccione la opción 'Close'.

Para seleccionar los campos simplemente haga un clic sobre su nombre en la representación gráfica de cada Tabla, aparecerá en forma enmarcada, si desea desmarcarla simplemente vuelva a hacer un clic sobre esta.

## Fichero de la función Tabla

La barra de útiles se presenta en la parte inferior de la ventana como tópicos de trabajo. Cada uno de estos tópicos corresponden a alguna parte de una sentencia SQL.



### Fichero de tabla

### Group (Agrupar)

Se refiere a la posibilidad de agrupar el resto de los campos por un campo particular, esto permite que todos los registros que tengan el campo por el que se esta agrupando con un mismo valor, este uno junto con el otro.

Haga un clic sobre un campo de la lista que aparece a mano izquierda y sin soltar el botón del ratón coloque en el espacio que hay en la mano derecha.

### Sort (Ordenar)

La presentación de los registros es ordenada por el valor de un campo definido, ya sea ascendente o descendente.

Haga un clic sobre un campo de la lista que aparece a mano izquierda y sin soltar el botón del ratón colóquelo en el espacio que hay en la mano derecha. Al lado de este nuevo elemento hay un cuadrado que si esta marcado indica que los registros serán ordenados en forma ascendente con respecto a los valores del campo.

### Where (Condicionar)

La presentación de los registros puede ser condicionada, a través de una(s) expresión(es) condicional(es), todas estas deben referirse al valor de un campo.

Column	Operator	Value	Logical
personas.rut	like	'12%'	And
personas.fecha_nac	>	:fech_nac	

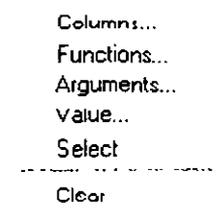
Cada columna se refiere a alguna parte de una expresión condicional.

Columna	Significado
<b>Column</b>	Campo a ser consultado, puede ser cualquier campo de las tablas que participan o bien una función que utilice un campo; inclusive un campo que no ha sido marcado para seleccionarlo.
<b>Operator</b>	Operador lógico, Los operadores lógicos son:
	<b>igualdad</b> =
	<b>menor</b> <
	<b>mayor</b> >
	<b>menor o igual</b> <=
	<b>mayor o igual</b> >=
	<b>diferencia</b> <>
	<b>entre</b> Between
	<b>Existe en expresión</b> Exist
	<b>en</b> In
	<b>Como (texto)</b> Like
	<b>Es</b> Is
<b>Value</b>	Valor de la expresión, ver la explicación mas adelante
<b>Logical</b>	Operador condicional, si la condición Where requiriera de múltiples expresiones condicionales deberán separarse dichas expresiones por un operador condicional
	Y inclusivo And
	O inclusivo Or

El valor de esta expresión puede ser:

- Podrá escribir el valor literal, siguiendo las siguientes reglas:
- Si fuese un valor numérico, simplemente escríbalo. **Ej.** 45; 6,7
- Si fuese un valor texto o fecha, deberá anteponerlo y prescederlo con una comilla simple (') **Ej.** 'Arturo Benitez'

Otro valor, para extraer otro valor, estando el cursor sobre la casilla del valor presione el botón derecho del ratón y se desplegar un listado de opciones.



Items	Descripción
Columns	Muestra un listado de todas las columnas que están en las tablas seleccionadas para el Origen de Datos.
Functions	Muestra un listado de las funciones que pueden ser utilizados.
Arguments	Muestra un listado de los argumentos definidos para este Origen de datos.
Value	Muestra un listado de todos los valores que están presentes en el campo, definido en la columna 'columns', en la Base de datos.
Select	Permite diseñar una sentencia select anidada, es igual que diseñar otro origen de datos. La consulta así definida deberá devolver un solo campo del mismo tipo que el definido en la casilla 'columns' y podrá ser un valor o múltiples valores ( un registro o múltiples registros).
Clear	Permite limpiar toda la sentencia definida, tanto la columna, el operador y el valor para la expresión.

### Compute (Cálculos)

Es posible definir cálculos, estos cálculos deben ser funciones agregadas de SQL. **Ej.** count(personas.rut)

### Having

La sentencia 'Where' permite definir los registros que serán seleccionados. Pero si la consulta tiene grupos, podría desear que solo algunos grupos sean seleccionados.

## Sintax (Sintaxis)

Este t3pico muestra la sintaxis SQL que origina la presentaci3n del Datawindows. Esta sentencia es utilizada por obtener los datos que ser3n mostrados en la presentaci3n del Datawindows.

## Par3metros

Par3metro es un valor que es traspasado a la consulta (Origen de Datos) para ser utilizada por esta como una variable. Para trabajar con par3metros se requiere de 2 pasos:

### Declarar Par3metro

Esto se realiza seleccionando la opci3n 'Retrieval Argument' del men3 'Design', aparecer3 una ventana donde podr3 escribir el(los) par3metro(s) que se quiera definir para este origen de datos.

Argument

Position	Name	Type
1	tech_nac	String

Los 3 columnas para definir un argumento son:

Nombre	Definici3n
<b>Position</b>	Indica su posici3n, dentro de la lista de argumentos.
<b>Name</b>	Debe escribirse un nombre, el que se le dar3 y ser3 utilizado dentro del Origen de Datos.
<b>Type</b>	Debe seleccionarse un tipo de dato, de los existentes.

Si se desea a1adir un nuevo par3metro al final de los ya existentes presione 'Add'. Si desea insertar un par3metro entre otros, presione 'Insert' y para eliminar un par3metro presione 'Delete', teniendo presente que el cursor este en alguna columna del par3metro a borrar.

## Utilizaci3n

Para utilizar un par3metro, ub3quese en la columna 'value' del t3pico 'where' de la caja de 3tiles. Presione el bot3n derecho del rat3n y seleccione la opci3n 'Arguments'; elija el par3metro a utilizar y presione el bot3n 'Paste'.

## Otras opciones

### Registros Distintos

Cuando se tiene un conjunto de registros en los cuales algún(os) campo(s) o todos inclusive, de los elegidos en la representación gráfica de la Tabla, se repitan y uno desea que todos los registros mostrados sean distintos, utilice la opción 'Distinct' del menú 'Design'.

Para terminar de trabajar con el 'Origen de Datos' y comenzar a trabajar con la 'Presentación' presione el botón . Si desea volver y realizar modificaciones al Origen de Datos estando en Presentación presione nuevamente el mismo botón.

### Presentación

La presentación es un conjunto de objetos que son puestos en alguna posición de un Área de edición que representa el fondo; este puede ser utilizado como si fuese una pagina de un documento a imprimir. Cada uno de los objetos y secciones que conforman la presentación tienen un nombre y poseen propiedades bien definidas. Para terminar salga de la ventana o presione el botón que aparece en la tabla secundaria .

### Secciones de la Presentación

Las secciones son áreas donde pueden ponerse los objetos. Cada sección esta delimitada por una barra horizontal, que tiene escrita el nombre de la sección y una flecha que apunta hacia el área que corresponde a dicha sección. Para cambiar el largo de una sección mantenga presionado el ratón sobre la barra horizontal de esta y córrala hacia arriba o abajo; inmediatamente se agrandara el área correspondiente.

Las secciones son las siguientes.

<b>Header</b> ↑
Mscard mscard
<b>Detail</b> ↑
<b>Summary</b> ↑
<b>Footer</b> ↑

Nombre	Descripción
<b>Header</b>	Esta sección se imprime en cada pagina que se origine al momento de imprimir, y corresponde a un encabezado de pagina.
<b>Detail</b>	Esta sección se imprime para cada registro que contenga la sentencia de Origen de Datos.
<b>Summary</b>	Esta sección es utilizada cuando existan calculos , que estén relacionados con campos del 'Origen de Datos' sean funciones agregadas de SQL.

<b>Footer</b>	Esta sección se imprimirá en cada pagina, y corresponde a un pie de pagina.
---------------	---

Adicional a estas secciones estándar, existen 2 que son definidas por el usuario al declarar un Grupo.

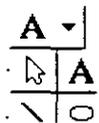
Nombre	Descripción
<b>Header Group</b>	Esta sección se imprime cada vez que cambie el valor para el cual fue definido el grupo. Cada Grupo tiene un numero que lo identifica.
<b>Trailer Group</b>	Esta sección es impresa cada vez que termina de listarse los registros del Grupo.

Para ver las propiedades de la hoja de la presentación, presione el botón derecho del ratón estando este en un área vacía de la presentación. Estas propiedades están divididas en tres tópicos

Tópicos	Descripción
<b>General</b>	Determina la unidad de medida para la presentación, e color de fondo de la presentación.
<b>Pointer</b>	El tipo de puntero del ratón que será utilizado cuando este pase por encima del Área vacía de la presentación
<b>Printer Specification</b>	Se definen las especificaciones al imprimir; como son e tipo de papel, la Orientación.

## Elementos

Existen múltiples elementos (objetos) que pueden ser anexados a la presentación; todos estos están listados en el botón  de la barra de herramientas 'PainterBar'



Para anexar un nuevo objeto a la presentación haga un clic sobre este en el listado y luego haga otro clic en el lugar (en la presentación) donde desea que aparezca.

Para mover un objeto, haga un clic sobre este y sin soltar el botón del ratón muévelo a otra ubicación; soltando el botón donde desea que se ubique el objeto. Para seleccionar varios objetos mantenga presionada la tecla 'Control' mientras los selecciona.

Para eliminar un(os) objetos selecciónelo(s) y presione la tecla 'Suprimir'.

Para ver las propiedades de un objeto; presione el botón .

Para modificar característico de presentación de varios a la vez:

- Selecciónelos.
- Utilice los botones de la barra secundaria (colores, justificación, etc.), no puede modificarlos presionando el botón de propiedades del objeto.



Los objetos definidos se explicaran a continuación:

Nombre	Icon	Descripción
<b>Text</b>		Este objeto es una etiqueta , muestra un texto pero no presenta una caja de texto para ser modificado directamente en tiempo de ejecución.Los tópicos de su Propiedades son:
<b>Picture</b>		Permite seleccionar un archivo de dibujo para ser mostrado en la presentación del Datawindows
<b>Line</b>		Permite poner esos objetos en la presentación
<b>Oval</b>		
<b>Rectangle</b>		
<b>Round Rectangle</b>		
<b>Column</b>		Permite añadir una columna de la sentencia SQL como si fuese un objeto más. Si es que accidentalmente borro el objeto que se relacionaba con dicha columna.
<b>Compute</b>		Permite que Defina funciones agregadas y expresiones condicionales. Todas estas deben presentar un valor por pantalla.
<b>Graph</b>		Permite insertar un gráfico, los datos para diseñar el gráfico serán alguna de las columnas definidas en el Origen de Datos.
<b>Nested Report</b>		Permite ingresar un subreporter en la presentación del Datawindows, este reporte debe ser un Datawindow previamente diseñado. Por ende podemos decir que este objeto permite crear un Datawindow subordinado.
<b>Ole</b>		Permite insertar un objeto Ole. Este objeto Ole puede ser uno Nuevo, un archivo ya existente o un objeto externo a PowerBuilder pero registrado en el sistema.

<b>Page field</b>  <b>Today</b>	  	Son objetos compute que poseen sentencias predefinidas, pueden ser modificados:  Page, muestra un mensaje automatizado de la pagina impresa y el total de estas.  Today, muestra un mensaje automatizado para presentar la fecha actual.
<b>Average</b> <b>Count</b> <b>Sum</b>	    	Estos son objetos que poseen sentencias predefinidas, pueden ser modificadas. Para hacer uso de esto, debe marcar un campo y luego seleccionar el objeto; este campo sera utilizado para sentencia automatizada.  Average, entrega el promedio de un campo numérico.  Count, entrega la cuenta de los registros.  Sum, entrega la suma de un campo numérico.

## Propiedades

Cada objeto posee propiedades, los que pueden ser vistos presionando el botón  o bien haciendo doble clic sobre el objeto. Para diferenciar los tópicos que conforman los atributos; existen 2 tipos de objetos.

### a) Elementales:

Son los objetos que no son computados, es decir la presentación de información no depende directamente de un calculo, condición o expresión definida como parte del mismo objeto. Los objetos Elementales son:

Text, Picture, Column, Line, Rectangle, Oval y Round Rectangle

### b) Computados y Subordinados:

Son los objetos son dependientes, en el primer caso de un calculo, condición o expresión definida en el mismo objeto, y en el segundo caso, son elementos que son externos a la presentación:

#### Computados:

Compute, Average, Count, Page Computed Field, Sum, Today.

*Subordinados:*

Graph, Nested Report, Ole.

### Atributos elementales

Los atributos están ordenados en tópicos, no todos los objetos poseen todos los tópicos, existen diferencias. Los tópicos son:

General | Font | Position | Pointer | Expressions |

Tópico	Descripción
<b>General</b>	Si fuese un objeto <i>Text</i> o un objeto <i>Column</i> , se puede definir el nombre del objeto, el tipo de borde y la alineación del texto dentro del ancho del objeto y el texto que se debiera mostrar. Si fuese un objeto <i>Picture</i> , se puede seleccionar el dibujo a mostrar con el botón 'Browse', la opción 'Original size' permite que se muestre el dibujo con su tamaño real. Si fuese un objeto <i>Line</i> , <i>Rectangle</i> , <i>Oval</i> o <i>Round Rectangle</i> , en la opción 'Line' puede definir el color del borde, en 'Fill' el color del relleno definido en 'Pattern', y finalmente el grosor de la línea en 'Line'.
<b>Font</b>	Si el objeto fuese <i>Text</i> o <i>Column</i> , puede determina la Fuente de texto, tamaño y efectos adicionales. En la opción 'Text Color' se elige el color del texto y en la opción 'BackGround' el color de fondo del objeto.
<b>Position</b>	Para todos los objetos Elementales, permite escribir las coordenadas donde se presentara el objeto. La opción 'Autosize' permite que se autoajuste el texto al ser modificado. La opción 'Resize' permite que pueda ajustarse el tamaño del objeto en tiempo de ejecución. La opción 'Moveable' permite que el objeto pueda ser movido de su ubicación original en tiempo de ejecución.
<b>Pointer</b>	Define el tipo de puntero del ratón que aparecerá cuando el ratón este sobre el objeto.
<b>Expresions</b>	Es un listado de los atributos modificables del objeto en tiempo de ejecución. Estos pueden tener una expresión; sea esta condicional. <b>Ej.</b> El atributo visible. Si yo deseo que este objeto no sea visible, el valor de este atributo debe ser 'Falso', debo escribir en la casilla contigua al atributo 0=1 (esta expresión es falsa). Por ende el objeto no será visible.

El objeto Column posee atributos especiales y que es necesario aclararlos aparte.

## Reglas de Presentación Extendidas

El objeto Column posee 3 tópicos dentro de sus atributos que son muy importantes. Todos ellos se trabajan igual que elementos "Estilos de Edición", "Formato de Presentación" y "Reglas de Validación" del Area de Base de Datos

| Edit | Format | Validation |

Tópico	Descripción
<b>Edit</b>	Permite seleccionar un estilo de presentación para los datos de la columna, o para los datos que podría tomar dicha columna
<b>Format</b>	Permite definir un formato de entrada de datos, esto es una regla para los caracteres (letras, numeros, simbolos) que pueden ser utilizados, inclusive su posición dentro del campo.
<b>Validation</b>	Permite definir una regla de validación. Si esta regla (condicional) devuelve 'Verdad' el dato entrado/seleccionado es aceptado.

### Atributos Objetos Computados y Subordinados

Los atributos están ordenados en tópicos, no todos los objetos poseen todos los tópicos, existen diferencias. Los tópicos que se listan a continuación son los tópicos distintivos de estos tipos de Objetos:

Tópico	Descripción
<b>General</b>	Si el objeto fuese <i>Compute</i> o bien un objetos compute que poseen sentencias predefinidas, en la caja inferior se debe escribir la expresión condicional. Para manejarla en mejor forma presione el botón 'More'; se presentara una ventana con todas las característico para el diseño de sentencias. El listado 'Functions' muestra las funciones que pueden utilizarse, haga un clic sobre una y esta se pegara en la caja de edición superior. El listado 'Columns' muestra los campos de puede utilizar para la expresión. El botón 'Verify' comprueba si la expresión condicional esta correctamente escrita.
<b>Select Report</b>	Solo para objetos <i>Nested Report</i> , permite seleccionar el Datawindow subordinado, de los ya diseñados.
<b>Criteria</b>	Solo para objetos <i>Nested Report</i> , permite definir un criterio para la selección de registros a mostrar en el Datawindow subordinado. Cada campo del datawindows subordinado aparece como una columna. En la casilla criterio correspondiente escriba un criterio. <b>Ej.</b> <34, 'Jose donoso' Si no existiera un operador relacional se entiende que solo muestre los registros que sean igual al valor ingresado. Cada fila indica otro criterio de presentación.

## Grupos

Los grupos son una manera de organizar la información al ser desplegada, es así como al indicar un grupo se debe indicar el campo por el cual se desea agrupar; esto significa que todos los registros cuyo valor para dicho campo sea igual serán mostrados uno a continuación del otro.

### Generar un Grupo

Para generar un nuevo grupo seleccione la opción 'Create Group' del menú 'Rows'. Aparecerá una ventana como la que sigue

Source Data	Columns
msgid	
msgtitle	

Mueva el campo que generara el grupo, desde la casilla 'Source Data' a la casilla del lado izquierdo, "Columns".

La opción 'New Page ...', significa que cada vez que cambie el valor del campo que genera el grupo se realizara un salto de pagina automático. La opción 'Reset Page ..', significa que cuando se liste un nuevo grupo el contador del numero de pagina volverá a cero.

General | Pointer | Definition | Sort Group | Expressions |

Tópico	Descripción
<b>Sort Group</b>	Permite definir el (los) campos por el cual serán ordenados los registros dentro del grupo. Para ello mueva el campo a la casilla que aparece al lado izquierdo
<b>General</b>	La opción 'Color' permite seleccionar el color que tendrá la cabecera del Grupo. La opción 'Height' permite determinar el alto de la cabecera del Grupo.
<b>Definition</b>	Permite modificar el campo que genera el grupo.

Cuando se genera un nuevo grupo se crean dos secciones nuevas 'Header Group' y 'Trailer Group'; las que fueron definidas previamente.

### Modificar Atributos Grupo

Para modificar alguno de los atributos que se definió al crear el grupo. Seleccione la opción 'Edit Group' del menú 'Rows'; aparecerá un número, este número es el que se ve en las barras diagonales que separan al grupo y que fue asignado automáticamente al generar el grupo, elija el número que corresponde al grupo que desea modificar. La ventana que aparecerá es la misma que apareció al generar el grupo, con todos sus tópicos; realice las modificaciones y presione el botón 'Ok' para terminar.

## Eliminar Grupo

Para eliminar algún grupo definido. Seleccione la opción 'Delete Group' del menú 'Rows'; aparecerá un número, este número es el que se ve en las barras diagonales que separan al grupo y que fue asignado automáticamente al generar el grupo, elija el número que corresponde al grupo que desea Eliminar. Cuando lo haga, la definición del grupo y todos los objetos que estaban en las secciones 'Header Group' y 'Trailer Group' del respectivo grupo serán eliminados.

## Otros

### Filtro (Filter)

Esta opción permite realizar una presentación condicional de registros. La expresión condicional se puede referir a cualquier campo de la consulta de Origen de Datos. Para hacer uso de esta seleccione la opción 'Filter' del menú 'Rows'. Aparecerá una ventana idéntica a la mostrada cuando se diseña una regla de validación para un objeto campo.

### Specify Filter

msgid > 12

Escrita la condición en la caja superior, para comprobar si esta correcta presione el botón 'Verify'. Utilice el listado 'Functions', para añadir una función a la expresión condicional, y el listado 'Columns', para añadir una columna a la expresión condicional, estas dos lo añaden en la posición actual del cursor.

### Presentación Preliminar:

Para hacer una presentación preliminar presione el boton <boton pres. Pre.> o elija la opción 'Preview' del menú 'Design'.

Estando en el modo "Actualización Datos Tabla":

- Si se desea añadir un nuevo registro al final de los ya existentes presione el botón .
- Para eliminar un registro presione el botón , se eliminara el registro donde este el cursor alcualmente.
- Para guardar los cambios en la Base de datos , presione el botón .
- Para obtener datos de la Base de datos, presione el botón .
- Para terminar el modo "Actualización ", presione el botón .
- Para moverse de un campo a otro presione la tecla 'Tab'. Para moverse de un registro a otro utilice las teclas cursor arriba() y cursor abajo() , para ir un registro antes y un registro despues respectivamente.

Para presentar una presentación previa de la impresión, seleccione la opción 'Print Preview' del menú 'File'. Después de esto. Para mostrar una regla de bordes, seleccione la opción 'Print Preview Rulers' del menú 'File', y para hacer un zoom, seleccione la opción 'Print Preview Zoom' del menú 'File'.

### Orden de Tabulación

El orden de tabulación indica a que campo ira el cursor cuando se presione la tecla Tab., el orden de tabulación es numérico ascendente. Para definir el orden de tabulación presione el botón  o bien elija la opción 'Tab Orden' del menú 'Design'. Durante la definición del orden de tabulación todas las demás operaciones quedan inactivas, para volver a la normalidad presione el mismo botón o seleccione la misma opción.

### Utilización DataWindows

El objeto DataWindows en si no es de ninguna utilidad, si no se vincula con un objeto Datawindow control. Este objeto, es visual y es posible de insertar directamente en una ventana; para que pueda ser utilizado por el usuario.

### Objeto DataWindow Control

El objeto Datawindow Control se selecciona del listado de objetos que se pueden insertar en una ventana, simplemente haga un clic sobre el icono . Luego haga un clic en la posición, en la ventana, donde desea poner el Datawindow control.

Haga doble clic sobre el objeto DataWindows que esta ahora en la ventana, o bien presione el botón , teniendo seleccionado el objeto, para ver las propiedades que posee.

A continuación se listan las propiedades particulares que el objeto posee.

Propiedad	Descripción
<b>Name</b>	Es el nombre que tendrá el objeto Datawindow Control.
<b>Datawindow</b>	Escriba el nombre del objeto Datawindow que será mostrado en este Datawindow control. Para seleccionar un Datawindow presione el botón "Browse".
<b>Title</b>	Escriba el texto que aparecerá en la barra de titulo, si esta marcada la opción 'TitleBar'.
<b>TitleBar</b>	Si esta opción esta marcada, el texto que este en la caja 'Title' se mostrara como una barra de titulo.
<b>Control Menú</b>	Si esta opción está marcada, permite que el Datawindow Control, sea un objeto flotante (pudiendo moverse) dentro de los limites de la ventana que lo contiene.
<b>Maximize box</b>	Permite que el Datawindow control, presente los botones Maximizar y Minimizar, respectivamente.

## Manejo de Datawindow Control

Para hacer uso del Datawindow Control en tiempo de ejecución es necesario indicar que objeto de transacción será utilizado por el Datawindow (ver el tema "Base de Datos") para presentar la información definida en el.

Esto se hace, con la siguiente función:

```
< nom Dw > .SetTransObject(< obj transac >)
```

Donde:

- < nom Dw > es el nombre del Datawindow Control.
- < obj transac > es el objeto de transacción, por defecto utilice SQLCA.

Esta sentencia debe escribirse antes de hacer uso del Datawindow Control, normalmente se pone en el Evento "Open" de la ventana que contiene al Datawindow Control.

Ej.

```
dw_inicio.SetTransObject( sqlca )
```

- Para obtener datos de la Base de Datos, y mostrarlos en el Datawindow Control, se utiliza la siguiente función:

Esto se hace, con la siguiente función:

```
< nom Dw > .Retrieve (< lista parametros >)
```

Donde:

- < nom Dw > es el nombre del Datawindow Control.
- < lista parametros >, si para el Origen de Datos' (del Datawindow), se definieron argumentos, deben listarse los valores a utilizar por cada uno en el mismo orden en que fueron definidos y separados entre sí por una coma. Los argumentos se escriben como literales, sin comillas para los argumentos de tipo texto.

Ej.

```
dw_nuevo.Retrieve( Contabilidad, sle_edad.text )
```

Para modificar en tiempo de ejecución el datawindow que esta vinculado al Datawindow Control, utilice la siguiente función:

`<nom Dw>.DataObject = '<nombre obj Dw>'`

Donde:

- `<nom Dw>` es el nombre del Datawindow Control.
- `< nombre obj Dw >`, es el nombre de un objeto Datawindow definido. Si el objeto Datawindow vinculado al Datawindow Control, es necesario nuevamente, indicar el objeto de transacción que se utilizara para este Datawindow y volver a ocupar la función **Retrieve**, para obtener los datos de esta nuevo Datawindow.

Ej.

`dw_nuevo.DataObject= 'd_otro'`

### Funciones para el manejo.

Añadir  
Eliminar  
Buscar

Filtrar Registros  
Actualizar Base de Datos

---

### Añadir

Para añadir un nuevo registro se utiliza la siguiente función:

`<nom Dw>.InsertRow(<pos>)`

Donde:

- `<nom Dw>` es el nombre del Datawindow Control.
- `<pos>` Es la posición, dentro del número de registros mostrados en el Datawindow Control, donde se insertara el nuevo registro. Si `<pos> = 0`, añade un registro al final de los ya existentes. El añadir un registro, para ser llenado los datos de los campos por el usuario, no implica que el foco se mueva a este nuevo registro.

Ej.

`dw_nuevo.InsertRow(1)`

### Eliminar

Para eliminar un registro debe conocer cual es la posición, dentro del número de registros mostrados en el Datawindow Control, del registro a eliminar.

Esto se hace, con la siguiente función:

`<nom Dw>.DeletedRow(<pos>)`

Donde:

- `<nom Dw>` es el nombre del Datawindow Control.
- `<pos>` es la posición del registro a eliminar. Si desea eliminar el registro donde esta el foco (cursor actualmente), utilice la siguiente función: `<nom Dw>.GetRow( )`

Ej.

`dw_nuevo.DeletedRow( dw_nuevo. GetRow( ) )`

## Buscar

Permite buscar un registro, dentro de los listados en el Datawindow Control, que coincida con la expresión definida. Para esto utilice la siguiente función:

`<nom Dw>.Find(<expre cond>)`

Donde:

- `<nom Dw>` es el nombre del Datawindow Control.
- `< expre cond >` es una expresión que se escribe igual que una expresión Where. Esta función devuelve la posición del registro que cumpla con la condición, o un cero si no encuentra ninguno.

Ej.

integer pos

`pos= dw_nuevo.Find("rut= '" + sle_rut.text + "'")`

## Filtrar Registros

Esto significa que del conjunto de registros que muestra el Datawindow Control en forma habitual, solo se mostraran los que cumplan con cierta condición. Quedando los otros invisibles, pero presentes en el Datawindow Control. Para esto posee de dos funciones, la primera para definir una filtro, y la segunda para ejecutar el filtrado definido.

- Definir Filtro: Si desea modificar el filtro definido en la "Presentación" del Datawindow, o crear un nuevo filtro, utilice la siguiente función:

*< nom Dw >.SetFilter(< expre cond >)*

Donde:

- *< nom Dw >* es el nombre del Datawindow Control.
- *< expre cond >* es una expresión que se escribe igual que una expresión Where.

Ejecutar Filtro: Para realizar el filtrado en el Datawindow Control, utilice la siguiente función:

*< nom Dw >.SetFilter( )*

Donde: *< nom Dw >* es el nombre del Datawindow Control.

Ej.

dw\_nuevo.**SetFilter**("Edad > 21")

dw\_nuevo. **SetFilter**( )

### **Actualizar Base de Datos**

Debe indicársele que traspase la información modificada/eliminada/ actualizada, desde el Datawindow Control a la Base de Datos.

Para esto utilice la siguiente función:

*< nom Dw >.Update( )*

Donde: *< nom Dw >* es el nombre del Datawindow Control.

Esta función devuelve un valor, -1 si un error ocurrió

Ej.

If dw\_nuevo. **Update**( ) <> -1 then Commit ;

Else

    Rollback ;

End If

### **Obs.**

La sentencia 'Commit', guarda los cambios permanentemente en la Base de Datos.

La sentencia 'Rollback' deshace todas las modificaciones hechas en la Base de Datos, esto siempre se debe hacer si se encuentra un error.

## Modificar Presentación

Los objetos que están contenidos en el Datawindow, y que fueron definidos en la "Presentación" del mismo, pueden ser manipulados, por la utilización de 2 funciones básicas:

- Describe
- Modify

---

### Describe

Esta función nos permite conocer el valor de alguna propiedad o atributo de un objeto, sección, característica del Datawindows. Su sintaxis es la siguiente:

*< nom Dw >.Describe(< expre consulta > )*

Donde:

- *< nom Dw >* es el nombre del Datawindow Control.
- *< expre consulta >*, es una expresión que debe contener el objeto, sección, o característica a consultar.

Ej.

boolean paso

paso= dw\_nuevo. **Describe**("t\_rut.visible")

### Modify

Esta función permite realizar modificaciones al valor de algún atributo de un objeto, sección, característica del Datawindows. Su sintaxis es la siguiente:

*< nom Dw >.Modify(< expre modificar > )*

Donde:

- *< nom Dw >* es el nombre del Datawindow Control.
- *< expre modificar >*, es una expresión que debe contener el objeto, sección, o característica a modificar.

Ej.

```
dw_nuevo.Modify("t_rut.visible = True")
```

## Imprimir DataWindow Control

Para imprimir el contenido del Datawindow utilice la siguiente función:

```
< nom Dw>.Print()
```

Donde: < nom Dw> es el nombre del Datawindow Control.

Para mostrar una caja de dialogo, donde seleccionar la impresora a utilizar y poder modificar parámetros de impresión. Utilice la siguiente función.

```
PrintSetup( )
```

## Presentación Preliminar

- Para presentar el datawindow con presentación preliminar, escriba la siguiente sentencia:

```
< nom Dw>.Modify("Datawindow.Print.Preview = Yes")
```

- Para presentar en el Datawindow, en modo presentación preliminar, una regla de bordes de impresión, escriba la siguiente sentencia:

```
< nom Dw>.Modify("Datawindow.Print.Preview.Rulers = Yes")
```

- Si desea hacer un Zoom a los datos del Datawindow, en modo presentación preliminar, escriba la siguiente sentencia:

```
< nom Dw>.Modify("Datawindow.Print.Preview.Zoom = '< Valor>")
```

Donde: < Valor> es el valor del Zoom a mostrar.

Ej.

```
< nom Dw>.Modify("Datawindow.Print.Preview.Zoom = '50")
```

```
dw_nuevo.Modify("Datawindow.Print.Preview = Yes")
```

```
< nom Dw>.Modify("Datawindow.Print.Preview.Rulers = Yes")
```

## Eventos

Como ya es de su conocimiento, los eventos son las acciones; que son realizadas por el usuario, o cuando se altera alguna propiedad/atributo del Objeto.

El Objeto Datawindow Control también posee eventos, los más comunes son los siguientes:

Evento	Descripción
<b>RowFocusChanged</b>	Sucede cuando el foco cambia de un registro a otro, no de un campo a otro dentro del mismo registro. También sucede cuando se obtienen datos (Retrieve) de la Base de Datos.
<b>ItemChanged</b>	Ocurre cuando el usuario un campo ha sido modificado y este campo pierde el foco ( presionar tecla <b>Enter</b> , tecla <b>Tab</b> , o moviéndose a otro campo cualquiera) También ocurre cuando se utiliza la función 'AcceptText' o la función 'Update'.

### SQL Incrustado

SQL (Structured Query Language), Lenguaje Estructurado de Consulta, es un conjunto de sentencias que son utilizadas para manipular el conjunto de datos de una Base de Datos relacional.

Las sentencias SQL de PowerBuilder difieren ligeramente de las sentencias SQL estandar. Estas se dicen incrustadas ya que las sentencias definidas aparecen como sentencias propias del lenguaje PowerScript, esto último permite que la aplicación pueda realizar estas operaciones de manejo de datos en forma tal que la Base de Datos no se da cuenta que es una aplicación externa la que esta realizando operaciones sobre los datos.

Existen dos tipos de sentencias SQL Incrustadas:

#### **a) No Cursores**

Estas sentencias realizan la misma operación sobre un conjunto de registros, es decir, cada registro no es individualizable.

#### **No Cursores**

Dentro de este tipo existen 4 sentencias:

<u>Select</u>	Asigna campos a variables (objetos) de programa.
<u>Insert</u>	Añade un nuevo registro a una Tabla especifica.
<u>Update</u>	Modifica los valores de campos de un(os) registros..
<u>Delete</u>	Borra el(los) registro(s).

Después de seleccionar el tipo de sentencia aparecera la ventana de selección de Tablas.



Cuando haya terminado de trabajar con la sentencia presione el boton , y la sentencia SQL se pegara en el Script en el lugar donde este el cursor.

Las definiremos ahora.

## Select

Cuando se desea seleccionar, tomar datos de algún campo de una(s) Tabla(s); asignandosele el valor de cada campo seleccionado a una variable definida en el programa o bien a algun atributo de un objeto cualquiera, se debe utilizar esta opción.

Apareceran en forma grafica la(s) Tabla(s) y tambien las relaciones existentes entre ellas.

Seleccione los campos, haciendo un clic sobre ellos, en la representación grafica de la Tabla.

En la parte inferior de la ventana aparecen los topicos de trabajo, estos topicos se trabajan igual que cuando se selecciona un 'Origen de Datos' para un Datawindows.

Para cada campo que se seleccione de la(s) Tabla(s), debe especificarse una variable a la que sera asignado el valor de dicho campo, para asignar esto presione el boton  o seleccione la opción 'Into Variable' del menú 'Design', al realizarlo aparecera una ventana donde podra asignar la variable del programa u atributo de un objeto a cada campo.

Selected Column	Pr
al_alumno.nombre	
al_alumno...	

Al lado izquierdo de cada campo esta una caja de texto donde debera escribirse el nombre de la variable del programa, "Program Variable". Tiene 2 maneras para escribirla, estando el cursor parpadeando en la casilla de texto especifica:

- Escribir el nombre de la variable, anteponiendole dos puntos (: ).
- Hacer un clic sobre alguno de los atributos de los objetos que estan presentes en el listado que aparece en la parte inferior de la ventana "Programs Variables"; en este caso no se requiere anteponer los dos puntos. ( : )

## Insert

La sentencia Insert se utiliza cuando se desea añadir un nuevo registro en una Tabla.

Aparecera la ventana "Valores a Columna", que permite definir la variable/valor que se utilizara para cada campo al momento de ingresar un nuevo registro. Esto se realiza como sigue:

Column Name	Value
rut	null

Para cada campo que tiene la Tabla debera asignarle un valor. Este valor podra ser literal (numero o texto), variable del programa o bien el atributo de un objeto (normalmente el atributo *text*); de donde se tomara el valor a insertar en dicho campo.

Al lado izquierdo de cada campo, 'Column Name', esta una caja de texto donde debera escribirse el valor a asignarle al campo, 'Value'. Tiene 3 maneras para escribirla, estando el cursor parpadeando en la casilla de texto especifica:

Podra escribir el valor literal, siguiendo las siguientes reglas:

Si fuese un valor numérico, simplemente escríbalo. **Ej.** 45; 6,7

Si fuese un valor texto o fecha, deberá anteponerlo y precederlo con una comilla simple (') **Ej.** 'Arturo Benitez'

Haciendo un clic sobre alguno de los atributos de los objetos que están presentes en el listado que aparece en la parte inferior de la ventana, estos objetos son los que están insertos en la ventana o menú donde se desea pegar la sentencia.

Escribir el nombre de la variable del programa, de donde sacara el valor, teniendo el cuidado de anteponerle dos puntos ( : )

Adicional a lo anterior podrá indicar que el campo sea nulo, presionando el botón 'Null'. El botón 'Select' permite definir otra sentencia Select, de donde se extraerá el valor a ingresar a este campo.

Para aquel campo que sea su valor defecto el de 'autoincrement', haga un clic sobre el y presione el botón 'Delete'; para que no sea considerado en el ingreso del nuevo registro.

- no podrá hacer uso de los tópicos de trabajo que aparecen en la parte inferior de la ventana, solo podrá ver la sintaxis de la sentencia (Syntax)

Para volver a definir los valores a ser asignados a los campos, presione el botón .

## Update

La sentencia Update permite modificar el valor de un(os) campo(s) de una Tabla.

Aparecerá la ventana "Valores a Columna", que permite definir la variable/valor que se utilizara para cada campo al momento de actualizar su valor. Esto se realiza como sigue:

Column Name                      Value  
• rut = null

Seleccione el(los) campos a modificar, esto se hace haciendo un clic sobre el nombre del campo, en la representación grafica de la Tabla, que aparece en la parte inferior de la ventana.

Para cada campo que este en el listado superior de la ventana, deberá indicar el valor. Este valor podrá ser literal (numero o texto), variable del programa o bien el atributo de un objeto (normalmente el atributo *text*); de donde se tomara el valor a modificar en dicho campo.

Siga las mismas indicaciones dadas para el ingreso de valores de la sentencia "Insert".

Adicional a lo anterior podrá indicar que el campo sea nulo, presionando el botón 'Null'. Para extraer un campo de la lista de campos a actualizar, presione el botón 'Delete'.

Tenga cuidado de definir una expresión condicional en el Topico 'Where', o de otro modo, las modificaciones tendrán lugar sobre todos los registros de la Tabla.

Para volver a definir los valores a ser asignados a los campos, presione el botón .

## Delete

La sentencia Delete permite borrar un(os) registros de una Tabla, inclusive a partir de valores de otras tablas; esto ultimo definido como una Subconsulta en el Tópico de Trabajo 'Where' de la definición.

Después de presionar 'Ok' en la ventana "Paste SQL" aparecerá la ventana de "Origen de Datos". Lo primero que deberá hacer es seleccionar la tabla en la que desea eliminar registros de estas. Aparecerá en forma grafica la Tabla.

En la parte inferior de la ventana aparecen solo el tópico 'Where'.

Tenga cuidado de definir una expresión condicional en el Tópico 'Where', o de otro modo, eliminara todos los registros de la Tabla.

## Aclaración Final

La sentencia es finalizada con un punto y coma (;), si esta utilizando un Objeto de Transacción distinto al estándar (*sqlca*), inmediatamente antes del punto y coma escriba lo siguiente:

**USING** <otro obj>

Donde <otro obj>, corresponde a otro objeto de transacción definido en la aplicación.

Esto es de mucha importancia por lo tanto no olvide hacerlo.

<lista de variables> es una lista de variables del programa separados por coma y siempre llevando delante de cada nombre de variable los dos puntos (: ). Estas variables pueden ser cualquiera, incluso atributos de un objeto, siempre que el dato traspasado sea del mismo dominio de la variable.

### **b) Cursores**

Estas se caracterizan porque no son sentencias estáticas, es decir; dentro de la misma sentencia es posible 'recorrerla', esto es, poder capturar los datos de un registro en particular dentro de una consulta y luego capturar los datos de otro registro dentro de la misma consulta. Permite operar cada registro en forma individual, dentro de un conjunto definido de estos.

Ej.

Definir un cursor para las personas mayores de 65 años.

Para trabajar en SQL-Incrustado, debe estar en un modulo de programa, presione el botón  y aparecerá una ventana; donde podrá seleccionar el tipo de sentencia (Cursor, No-Cursor) y cual específicamente, y luego presione el botón 'Ok'.

### **Cursores**

Estas sentencias permiten definir un conjunto de campos y registros, sobre los cuales se podrá. Modificar valores, Eliminar registros y moverse dentro de los registros que cumplan con las restricciones definidas para la sentencia.

La mayor ventaja es que la sentencia definida es conservada en una variable, de tipo cursor, esta puede ser utilizada dentro del rango definido al declararla (Global, Shared, Instance, Local)

Para trabajar con cursores se definen 3 pasos:

- Declarar el Cursor, esto se hace estando en un modulo ejecutable de programa o bien en alguna casilla de declaración de variables.
  - Abrir el Cursor, esta operación inicializa la sentencia SQL definida en el paso anterior.
  - Trabajar con el cursor, esto se refiere a las acciones de Navegar, Actualizar y Eliminar.
  - Cerrar el Cursor, esto libera la memoria que utiliza la variable declarada como cursor.
- Se puede volver a Abrir el Cursor, si fuese necesario

Es una buena política comprobar el valor del `sqlca.sqlcode`, para ver si la operación realizada fue satisfactoria, al abrir, trabajar o cerrar el cursor.

### **Declarar el Cursor**

Para declarar el cursor. Si esta en un modulo ejecutable de programa presione el botón  y elija la opción 'Declare', si esta en alguna de las casillas de declaración de variables/constantes haga doble clic sobre el icono que dice 'Cursor', en ambos casos se continua como sigue.

Aparecerá la ventana "Selección de Tabla", solo las tablas que se seleccionen del listado será posible de manipular. Presione el botón 'Open' para que aparezcan las tablas seleccionadas, o presione el botón 'Cancel' para cancelar la operación. La opción 'System Tables' permite seleccionar las Tablas del catalogo de la Base de Datos; estas Tablas son utilizadas para la Administración de la Base de Datos. Las Tablas seleccionadas aparecen en la ventana en una forma grafica y también las relaciones existentes.

Seleccione los campos, haciendo un clic sobre ellos, en la representación grafica de la Tabla.

En la parte inferior de la ventana aparecen los tópicos de trabajo, estos tópicos se trabajan igual que cuando se selecciona el 'Origen de Datos' para un DataWindows.

Después de definir la sentencia, cierre la ventana, al hacerlo se preguntara el nombre que tendrá el cursor, presione el botón 'Ok', y la sentencia se pegara donde este el cursor en la ventana origen.

La sentencia es finalizada con un punto y coma (;), si esta utilizando un Objeto de Transacción distinto al estándar (`sqlca`), inmediatamente antes del punto y coma escriba lo siguiente:

**USING** <otro obj>

Donde <otro obj>, corresponde a otro objeto de transacción definido en la aplicación.

## Abrir el Cursor

Esta operación se realiza en un modulo de código ejecutable de programa. Para abrir un cursor, la sentencia es la siguiente:

```
OPEN <nombre cursor>;
```

Donde <nombre cursor> es el nombre que se le asigno al cursor al momento de declararlo. Recuerde que solo puede utilizar los cursores que estén definidos(Globales, Shared, Instance, Local) para su utilización en el rango que abarque el modulo en uso.

## Trabajar con el cursor

El trabajar con el cursor se refiere ha hacer uso de este para algún proceso. Es así como se puede obtener Datos de los registros que cumplan con las restricciones definidas al declarar el cursor. Tambien se pueden eliminar el registro donde este actualmente, de la lista de registros que cumplan con las restricciones definidas al declarar el cursor. Por último, tambien se puede actualizar del registro actual, el(los) campo(s) que forman parte de la declaración del cursor.

A continuación definiremos cada una de estas operaciones:

### Obtener Datos

Para obtener los datos de un registro se utiliza la siguiente sentencia:

```
FETCH <operador naveg> <nombre cursor> INTO <lista variables> ;
```

Donde:

- <nombre cursor> es el nombre del cursor de donde se desea obtener los datos. Si es la primera sentencia después de haber abierto el cursor, se obtendrán los datos de el primer registro que cumpla la condición fijada en la declaración del cursor, ya que el indicador de posición del registro actual esta en 0. Si se vuelve a repetir la misma sentencia se obtendrá el segundo registro, y así consecutivamente.
- <operador naveg> es una palabra reservada que se utiliza para indicar de donde, dentro de los registros que cumplan la declaración del cursor, se obtendrán los datos para llenar la <lista de variables>. Esta palabra reservada es optativa, ya que por defecto asume NEXT. Las palabras reservadas son las siguientes:

### Actualizar Datos

**Si se desea eliminar el registro actual del Cursor, utilice la siguiente sentencia:**

```
DELETE FROM <nom tabla> WHERE CURRENT OF <cur name> ;
```

Donde:

- *< nom tabla >*, es el nombre de la Tabla a eliminar el registro.
- *< cur name >*, es el nombre del cursor.

Si se desea modificar el valor de un campo, de los seleccionados cuando se definió el cursor, utilice la siguiente sentencia:

**UPDATE** *< nom tabla >* **SET** *< cols >* **WHERE CURRENT OF** *< cur >* ;

Operador	Descripción
<b>Next</b>	Indica que los datos serán obtenidos de siguiente registro, de los que cumplan con la declaración del Cursor.
<b>Prior</b>	Indica que los datos serán obtenidos de registro anterior.
<b>First</b>	Indica que los datos serán obtenidos de primer registro de los que cumplan con la declaración del Cursor.
<b>Last</b>	Indica que los datos serán obtenidos de último registro de los que cumplan con la declaración del Cursor.

Donde:

- *< nom tabla >*, es el nombre de la Tabla a modificarse.
- *< cols >*, es un listado de los campos, separados por coma. Nombre del campo, signo igual (=) y el valor a asignarse.
- *< cur >*, nombre del cursor

### Cerrar el Cursor

Para cerrar el cursor utilice la siguiente sentencia:

**CLOSE** *< nombre cursor >* ;

Donde: *< nombre cursor >*

El cerrar el cursor, no implica que se pueda volver a abrirlo en el futuro.

### Proyecto

Un proyecto es la definición del archivo ejecutable de la aplicación. Para trabajar en esta área presione el botón . Seleccione de la lista el proyecto a seguir trabajando. o presione

Seleccione el botón 'New', para comenzar uno nuevo, primero deberá definir el nombre y ruta del archivo ejecutable.

Aparecerá una ventana, que esta dividida en 2 secciones:

- **Definición Ejecutable**, permite definir las características que tendrá el archivo ejecutable. Las opciones "Generación del Código", permiten definir el formato del archivo ejecutable y el nivel de optimización.
- **Listado de Librerías**, deben aparecer todas las librerías que conforman la aplicación, de otro modo haga las modificaciones en la aplicación correspondiente.

La casilla 'PBD', permite declarar que se cree un archivo de Librería dinámica de PowerBuilder, escribiendo la ubicación y el nombre del archivo correspondiente.

Para generar el archivo ejecutable presione el botón .

## ¿Qué es PowerBuilder?

- PowerBuilder es un ambiente para desarrollar aplicaciones graficas cliente/servidor de alto desempeño para la familia Windows. PowerBuilder combina una interface gráfica intuitiva con un poderoso lenguaje de programación orientado a objetos (PowerScript). El front-end de PowerBuilder puede interactuar con la mayoría de DBMS basados en ODBC.

## Conceptos de Programación Orientada a Objetos

- **Clase** descripción abstracta de un grupo de objetos.
- **Objeto** instancia de una clase.
- **Atributos** características de un objeto.
- **Métodos** acciones que puede realizar un objeto.
- **Eventos** sucesos a los cuales reacciona un objeto.
- **Script** espacio en el cual se puede introducir código PowerScript.
- **Encapsulamiento** propiedad de un lenguaje que permite asegurar que el contenido de la información de un objeto está oculta al mundo exterior.
- **Herencia** define una relación entre clases, en donde una clase comparte la estructura o comportamiento definido en una o más clases (herencia simple y herencia múltiple, respectivamente).

## **Tipos de objetos en PowerBuilder**

En PowerBuilder todo parte de la clase llamada Powerobject.

- **Visuales** (graphicobject)
- **No visuales** (nonvisualobject)

A partir de éstas clases base, se pueden desarrollar las clases de usuario.

## **Barras de herramientas**

- **PowerBar** barra que se despliega cuando se inicia una sesión en PowerBuilder, es un parte principal de control para construir aplicaciones.
- **PainterBar** barra que se abre al ejecutar alguna aplicación de la powerbar para hacer tareas específicas de dicha aplicación.
- **StyleBar** barra para cambiar la presentación de un texto o la alineación de componentes

## Tipos de ventanas

- **Main** ventana independiente que actúa de forma independiente con el resto de las ventanas.
- **PopUp** ventana que es abierta desde otra ventana, que en la mayoría de los casos llegan a ser padres de las ventanas popup.
- **Child** ventana que es siempre abierta desde el interior de una ventana Main o Popup, que llegan a ser padres de la ventana hijo. *Una ventana hijo existe solo dentro de una ventana padre.* No puede tener menú y nunca se consideran como ventanas activas. La ventana hijo se cierra cuando se cierra la ventana que es padre.
- **Response** Las ventanas de respuesta solicitan información desde el usuario. Las ventanas de respuesta son de modo aplicación(application modal). Esto es, cuando una ventana de respuesta es desplegada, por tanto esta ventana se hace activa y las demás ventanas de la aplicación no son accesibles mientras el usuarios no responda a la ventana de respuesta.
- **MDI frame** es un marco de ventana en el que se puede abrir múltiples ventanas como documentos o Sheets(hojas) y moverse entre las hojas(sheets).
- **MDI frame con MicroHelp** agrega una barra de microayuda en la parte baja de la ventana MDI.

## Control

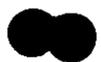
- Son objetos ubicados dentro de una ventana.

## Extensiones de archivos generados en PowerBuilder

- **.pbl** archivo de librería.
- **.pbd** archivo de librería compilada.
- **.exe** archivo ejecutable con ligas a los archivos pbd.
- **.srd** archivo generado cuando se exporta un datawindow.
- **.sra** archivo generado cuando se exporta un objeto aplicación.
- **.srf** archivo generado cuando se exporta una función.
- **.srm** archivo generado cuando se exporta un menú.
- **.srj** archivo generado cuando se exporta un proyecto.
- **.srs** archivo generado cuando se exporta una estructura.
- **.srw** archivo generado cuando se exporta una ventana.

## PowerScript

- Lenguaje de programación que utiliza PowerBuilder



## Tipos de datos

- Integer Partícula i.
- String Partícula s.
- Date Partícula d.
- Long Partícula l.
- Boolean Partícula b.
- Decimal Partícula c.

## Variables

### •Ámbito

- Local Partícula l.
- Global Partícula g.
- Shared Partícula s.
- Instance Partícula i.

### •Declaración

Tipo de dato Nombre de la variable

El nombre de la variable se compone de la siguiente manera:

PartículaÁmbitoPartículaTipodato\_NombreAutodocumentado

## Constantes

- Se declaran en mayúsculas, se le antepone la palabra *constant* y se aplican las mismas reglas que en la variables.

constant PartículaÁmbitoPartículaTipodato\_NombreAutodocumentado

## Operadores

- **Matemáticos** + - \* /

- **Relacionales**

= Igual

> Mayor que

< Menor que

<> Diferente

>= Mayor o igual que

<= Menor o igual que

- **Lógicos**

NOT Negación lógica

AND And lógico

OR Or lógico

## Precedencia

( )	Agrupación
+, -	Signos positivo y negativo para un número
^	Exponente
*, /	Multiplicación, división
+, -	Adición y sustracción; concatenación de cadenas
=, >, <, <=, >=, <>	Operadores relacionales
NOT	Negación
AND	And lógico
OR	Or lógico

## Flujos de control

- **Condicional**

**If.. Then..else**

**Choose..case**

- **Iterativos**

**For..next..step**

**Do..until..loop While..loop Exit**

Palabras reservadas

This

Parent

Painter de funciones

**Declaración**

**Parámetros**

**Paso de parámetros**

**Valor de retorno**

**Script**

Painter de sql incrustado

Painter de flujos de control

Compilador

Buscador

Comentarios



Estructuras

• **Definición**

• **Uso**

Menús

Librerías

Base de datos

• **Tablas**

• **Clave foránea**

• **Campos**

• **Índice**

• **Clave primaria**

• **Inserción de datos en la tabla**

Objeto transacción

• Es el área de comunicación entre el script y la base de datos. PowerBuilder define este tipo de objeto para facilitar la comunicación con la base de datos desde el código de programación.

• SQLCA es el objeto de acceso por defecto, cada vez que se utiliza una sentencia SQL dentro de un script.

1950  
1951

1952

1953  
1954  
1955

1956

1957  
1958

1959  
1960  
1961

1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050

1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050

Objeto datawindow

- Tipos
- Presentación
- Grupos

- Cálculo por grupos
- Filtros
- Tabulador

Control datawindow

- Funciones
- Describe

- Modify

SQL incrustado

- Cursores
- Select
- Insert

- Update
- Delete

