

Manual Instructivo

Power Builder V. 9.0

Prof. César Vásquez Pando

www.cevapan.net.tc

Huaraz, Abril de 2005

CAPITULO I

INTRODUCCION

Power Builder es un software de Aplicación que se utiliza para desarrollar aplicaciones de manera rápida y usa tecnologías RAD(Rapid Application Development (desarrollo rápido de aplicaciones) como cliente/servidor y las iniciativas orientadas al web.

Power Builder se usa como herramienta de desarrollo la cual utiliza y manipula datos de una o varias bases de datos a la vez de diferentes fabricantes de programas de gestión de bases de datos relacionales. Para ello debe conectarse a las base de datos utilizando una conexión para cada base de datos. En Power Builder se pueden crear aplicaciones no sólo para windows, sino también para UNIX y Macintosh. El propietario fabricante de Power Builder es la compañía SBASE® Inc. ..

VENTANAS (WINDOW)

Las ventanas (windows) son objetos de Power Builder que se usan como contenedores de objetos con la finalidad de implementar interfaces gráficas de usuario enriquecidas y que soportan la programación orientada a eventos y a su vez la programación orientada a objetos.

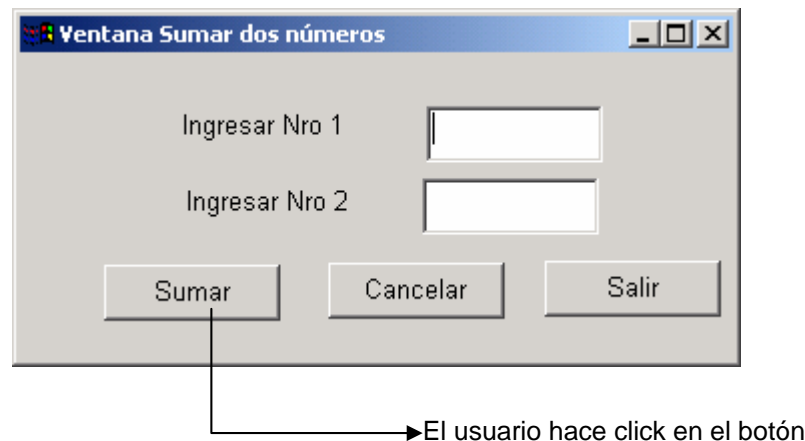
EVENTOS.

Las acciones que realice el usuario sobre el programa se llama EVENTOS, son eventos típicos el click sobre un botón, el hacer doble click sobre el nombre de un archivo para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón.

PROGRAMACION ORIENTADA A EVENTOS.

Consiste en presentar al usuario una interfaz gráfica y esperar a que él interactue con la misma, al hacerlo se desencadenarán; es decir ocurrirán sucesos o situaciones que deseamos controlar, por ejemplo: el usuario hace click en un botón.

Ejemplo: Ventana de Sumar dos números.



Los lenguajes visuales orientados al evento y con manejo de componentes dan al usuario que no cuentan con mucha experiencia en desarrollo, la posibilidad de construir sus propias aplicaciones utilizando interfaces gráficas sobre la base de ocurrencia de eventos.

Para soportar este tipo de desarrollo interactúan dos tipos de herramientas, una que permita realizar diseños gráficos y un lenguaje de alto nivel que permita codificar los eventos. Con dichas herramientas es posible desarrollar cualquier tipo de aplicaciones basadas en el entorno. Power Builder Soporta ampliamente estos requerimientos.

PROGRAMACION ORIENTADA A OBJETOS.

Es una técnica de programación que aumenta la velocidad de desarrollo de los programas y hace que su mantenimiento sea más fácil al volver a utilizar "objetos" que tienen comportamientos, características y relaciones asociadas con el programa. Los objetos son organizados en grupos que estén disponibles para la creación y mantenimiento de aplicaciones.

La programación orientada a procedimientos clásica presenta ciertos problemas que han ido haciéndose cada vez mas graves, a medida que se construyan aplicaciones y sistemas informáticos más complejos, entre los que destacan los siguientes:

Es difícil modificar y extender los programas, pues suele haber datos compartidos por varios subprogramas, que introducen interacciones ocultas entre ellos.

Es difícil mantener los programas. Casi todos los sistemas informáticos grandes tienen errores ocultos, que surgen a la luz hasta después de muchas horas de funcionamiento.

Es difícil reutilizar los programas. Es prácticamente imposible aprovechar en una aplicación nueva las subrutinas que se diseñaron para otra.

La programación orientada a objetos (OOP, por las siglas ingles Object-Oriented Programming) es una nueva forma de programar que proliferó a partir de los años ochenta y trata de encontrar solución a estos problemas utilizando los siguientes conceptos:

OBJETOS.

Entidades complejas provistas de datos(propiedades, atributos) y comportamiento (funcionalidad, programas, métodos). Corresponden a los objetos reales del mundo que nos rodea.

CLASES.

Conjunto de objetos comparten propiedades y comportamientos.

HERENCIA.

Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

ENCAPSULAMIENTO.

Cada objeto está aislado del exterior, es un módulo natural, y la aplicación entera se reduce a un agregado o rompecabezas de objetos. El aislamiento protege a los datos asociados a un objeto contra su modificación por quien no tenga derecho a acceder a ellos, eliminando efectos secundarios e interacciones.

POLIMORFISMO.

EL polimorfismo permite que una misma operación pueda llevarse acabo de forma diferente en clases diferentes, dependiendo de la referencia del objeto y el número de argumentos. Por ejemplo la operación mover, es distinta para una pieza de ajedrez que para una ficha de damas,

pero ambos objetos pueden ser movidos. Una operación es una acción o transformación que realiza o padece un objeto.

CARACTERISTICAS BASICAS DE POWER BUILDER

Power Builder es un software de aplicación para el desarrollador de aplicaciones, cuyas características básicas son:

- Soporta una gran variedad de sistemas de gestión de base de datos, tales como: Sybase, Informix, Oracle, Watcom, entre otras.
- Tiene capacidad de acceder a información de múltiples bases de datos y mostrar esa información en una única ventana.
- Se trabaja en ambiente cliente – servidor.
- Posee un objeto inteligente llamado Datawindow que realiza directamente la interfaz con la base de datos, sin requerir que el programador conozca SQL.
- Capacidad de utilizar sentencias SQL combinadas en el código.
- Se puede trabajar en múltiples plataformas, ya que soporta diferentes sistemas operativos y posee drivers nativos para las bases de datos más comerciales.
- La creación de aplicaciones es sencilla.
- Se realiza poca programación.
- Permite usar ventanas, botones y todas las herramientas que presenta el windows facilitando su manejo.
- Para construir la aplicación se utilizan painters (pintores), allí se definen las propiedades de los objetos y se agregan los controles.
- Se puede trabajar con múltiples ventanas.

NUEVAS CARACTERISTICAS.

Power Builder 9.0, en su más recientes versión es más rápido que nunca, siendo su entorno de desarrollo integrado(IDE) el más rápido que existe comparándolo a todas las herramientas RAD y lenguajes de programación.

Continúa soportando las plataformas .NET y J2EE.

Permanece la integración con Sybase PowerDesigner y es más sencillo ahora el rediseño de arquitecturas 4GL de aplicaciones para nuevos ambientes.

Se construye fácilmente aplicaciones web, poniendo RAD dentro de los JavaServer (™ Pages)

Es fácil construir y consumir servicios web de variada naturaleza. Abstrae la complejidad de XML

Le permite construir interfaces sofisticadas para EJB que residen en algún servidor de aplicaciones J2EE.

FUNCIONALIDADES.

- Un entorno de desarrollo profesional.
- Orientación a objetos.
- Rapidez de aprendizaje y desarrollo.
- Generación automática de código SQL.
- Acceso a las aplicaciones de windows.
- Una herramienta de desarrollo asequible y rentable en explotación.
- Un entorno abierto de desarrollo que dispone de interfaces inteligentes a otras tecnologías Cliente/Servidor.
- Abierto a la mayoría de Base de Datos del mercado.
- Abierto a diferentes herramientas CASE.
- Abierto a Librerías de objetos y control de versiones.
- Abierto a aplicaciones Windows: DLL, DDE, OLE.

CAPITULO II

CONCEPTOS Y TERMINOS

En esta sección describiré los conceptos y términos básicos que se requieren conocer antes de iniciar a trabajar con Power Builder, para el desarrollo de aplicaciones y componentes.

WORKSPACES Y TARGETS

El significado de workspace es espacio de trabajo y el significado de target es destino

En Power Builder, se trabaja con uno o más targets en un workspace. Se pueden agregar tantos targets como se necesite, abrir y editar objetos en múltiples targets, y construir y distribuir varios targets a la vez. Un target de Power Builder puede ser de dos tipos:

POWER SCRIPT TARGET.

Una aplicación ejecutable cliente/servidor o una aplicación ejecutable multicapa o un componente de servidor. La mayor parte de éste manual instructivo esta dedicado a las tareas de desarrollo de aplicaciones usando PowerScript

WEB TARGET

Una aplicación Web que puede contener todos los elementos que se necesite para construir un sitio web -archivos HTML, scripts, imágenes, componentes que se descargan – o una aplicación JavaServer Pages (JSP).un Web target además contiene configuraciones para construir opciones, realizar conexiones a base de datos y distribución.

La capacidad de crear Web Targets esta disponible únicamente en la versión empresarial de Power Builder (Enterprise Edition)

OBJETOS. Una aplicación es una colección de objetos. Power Builder provee de diversos tipos de objetos, inclusive objetos gráficos tales como ventanas, menús, botones y los objetos no visuales tales como los datastore, exception y objetos de transacción. En un Web Target, el objeto con el que se trabaje incluye páginas HTML, imágenes y hojas de estilo.

OBJETOS DATAWINDOW.

Las aplicaciones que se construyen giran alrededor de la organización de los datos. Con Power Builder se puede definir objetos DataWindow para recuperar, mostrar y para las operaciones de actualización de datos.

LIBRERIAS DE POWERBUILDER.

Todos los objetos o controles que se crean en Power Builder se guardan en una o más librerías asociadas con la aplicación (que están formadas por uno o más de dichos objetos y controles que estarán agrupados en archivos de extensión PBL) cuando se ejecutan las aplicaciones, Power Builder recupera los objetos de las librerías Pintores y editores.

Algunas herramientas que se usen para editar o crear objetos y controles se les llama Pintores. Por ejemplo, se puede crear una ventana en el pintor ventana. Dentro del pintor de ventanas se puede definir las propiedades de las mismas, agregar controles tales como botones y etiquetas, crear miniprogramas (scripts) usando Power Script para la ventana y para los controles que la misma contenga con el fin de instrumentar la funcionalidad que se desee obtener.

Power Builder proporciona pintores para ventanas (windows), menús, DataWindow objetos, objetos visuales y no visuales definidos por el usuario, funciones, estructuras, base de datos, data pipelines, y la aplicación en sí. Para cada uno de estos tipos de objetos, habrá también un editor de código en el que se puede modificar el código directamente.

Para webs, existen editores especializados para HTML y archivos JSP, frames sets, y hojas de estilo.

POWER SCRIPT

Es el lenguaje que se usa para desarrollar aplicaciones usando Power Builder, es decir está conformado por una serie de procedimientos, funciones, objetos (estos a su vez son no visibles pero poseen funciones y procedimientos propios denominados métodos o servicios y datos con

características propias los que se denominan propiedades o atributos), controles (que son visuales y son similares a los objetos en el sentido que se poseen también sus propios métodos y propiedades particulares)

Power Script soporta todas las características de la orientación a objetos (encapsulamiento, herencia y polimorfismo) SQL incrustado y excepciones (a partir de la versión 8).

EVENTOS Y SCRIPTS

Las aplicaciones en Power Script son orientadas a eventos o manejados por eventos: los usuarios controlan el flujo de la misma por las acciones o interacciones que ellos hacen. Cuando un usuario hace click en un botón, elige un elemento de un menú, o ingresa datos dentro de un cuadro de texto, un evento es desencadenado (es decir que se controla su ocurrencia mediante el manejador de windows así es posible para Power Builder verificar si existe un mini – programa (script) que especifique el procesamiento que debe llevarse a cabo cada vez que el evento es desencadenado.

Por ejemplo, los botones tienen el evento click. Se puede escribir un mini-programa para el evento click del botón que realice unas tareas específicas cada vez que el usuario de click en el botón. Similarmente, para cuando se edita los datos que muestran los controles existe el evento Modified, el cual es desencadenado cada vez que el usuario cambia un valor en el control.

Puede escribirse mini-programas (scripts) usando Power Script, el lenguaje Power Builder, en una vista de Script en el pintor para el objeto en que se está trabajando.

Los mini programas consisten en funciones de Power Script, expresiones y sentencias que son las que se ejecutan -el procesamiento- en respuesta a un evento. El mini programa para un evento click de un botón puede recuperar y mostrar información de una base de datos; el mini programa script para un control edit en el evento modified puede evaluar los datos y ejecutar procesamiento basado en dichos datos.

Los scripts pueden desencadenar eventos, por ejemplo, el script para un evento click de un botón puede abrir otra ventana, lo que a su vez desencadena el evento Open en dicha ventana.

FUNCIONES

Power Script proporciona un muy variado grupo de funciones pre construidas que se usan para actuar sobre los objetos y controles de una aplicación. Hay funciones para abrir una ventana, cerrar la misma, habilitar un botón, recuperar datos, actualizar una base de datos así por el estilo. Se puede además definir funciones propias para que procesen de manera particular una aplicación.

PROPIEDADES.

Todos los objetos y controles en un target tienen propiedades, muchos de los cuales puede configurarse conforme desarrolle la aplicación. Por ejemplo, puede indicar una etiqueta para un botón indicándola en la propiedad TEXT del mismo. Pueda establecerla en el pintor de ventanas o dinámicamente mediante un mini-programa dentro de un evento.

EXTENSIONES DE POWER BUILDER.

Puede extender la funcionalidad de un objeto añadiéndole nuevas propiedades (variables de instancia) y nuevas funciones y eventos.

CAPITULO III

DISEÑO DE LA INTERFAZ DE PROGRAMACION VISUAL EN POWER BUILDER

Una aplicación típica que interactúa con un usuario, tal como una hoja de cálculo sobre una computadora personal o un sistema de contabilidad sobre una mainframe, consiste de tres elementos: la presentación, la lógica de la aplicación y los datos.

La presentación es el conjunto de elementos que muestra la aplicación (ventanas, controles, menús) y está enfocada a la interacción con el usuario.

La lógica de la aplicación ejecuta los procesos y determina el flujo de la aplicación.

Los datos son los elementos que procesados proveerán de información al usuario.

Respecto a la configuración y despliegue de la aplicación, los desarrolladores tienen distintas opciones: aplicaciones de escritorio monolíticas corriendo sobre una computadora, aplicaciones multiusuarios corriendo sobre una mainframe, o aplicaciones cliente/servidor.

QUE ES UN CLIENTE/SERVIDOR.

Hemos mencionado tres elementos que conforman una aplicación: la presentación. La lógica de la aplicación, y los datos. Los desarrolladores tienen la posibilidad de separar la presentación de la lógica de la aplicación y los datos o también de separar los tres elementos creando de esta manera aplicaciones de dos capas, o aplicaciones de tres capas. La separación de los elementos en capas no implica necesariamente una separación física en la red de computadoras.

La manera como se diseña la separación en capas de los elementos de la aplicación define los diferentes modelos cliente/servidor:

Cliente Inteligente

Servidor Inteligente

Cliente/servidor en tres capas.

Internet.

LA APLICACIÓN EN POWER BUILDER

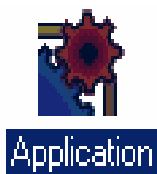
El desarrollo de una aplicación en Power Builder es un proceso visual en el que el diseño de las interfaces gráficas de usuario es similar al armado de un rompecabezas en base a piezas prefabricadas llamadas objetos. Lo que el desarrollador debe definir mediante la programación es cómo estos objetos responderán a las interacciones del usuario (programación dirigida por eventos). Power Builder es un entorno de desarrollo de aplicaciones para interfaces gráficas bajo una arquitectura CLIENTE/SERVIDOR.

LOS OBJETOS POWER BUILDER

Una aplicación Power Builder se construye en base a un grupo determinado de objetos del conjunto de objetos principales Power Builder:

- Application (Objeto aplicación)
- Window (Objeto ventana)
- DataWindows (Objeto DataWindow)
- Menú (Objeto Menú)
- Global Function (Objeto función)
- Query (Objeto Consulta)
- Structure (Objeto Estructura)
- User Object (Objeto Definido por el usuario)
- Project (Objeto proyecto)

APPLICATION



Una aplicación es una serie de actividades relacionadas, tales como actividades de recursos humanos o actividades referidas a la contabilidad de una organización. El objeto Aplicación es el punto de entrada a una colección

organizada de ventanas y otros objetos que permiten llevar a cabo dichas actividades. Es un objeto no visual que mantiene los valores por defecto para varias características de la aplicación.

WINDOW



El objeto Ventana es aquel que proporciona el componente principal de la interfase entre el usuario y la aplicación PowerBuilder. Una ventana puede mostrar información. Solicitar información al usuario, y responder a una acción del teclado o el ratón. Un objeto ventana tiene controles.

Los controles son objetos que se colocan en la ventana y que permiten al usuario interactuar con la aplicación

MENU



Un Menú es un objeto visual que proporciona al usuario una lista de comandos, opciones o modos alternativos de realizar una tarea.

DATAWINDOW



Un objeto DataWindow permite a la aplicación visualizar o capturar datos. Combina el acceso inteligente a la data con la interfase de usuario.

CAPITULO IV

Aplicación

Las aplicaciones son implementadas a través de objetos ***Application***, los que presentan propiedades, evento y funciones (métodos) que serán revisados. Una aplicación es un objeto no visual, desde el que se invocan a las interfaces (ventanas) que serán parte de una aplicación además de poder declarar variables globales, usar objetos de usuario, etc.

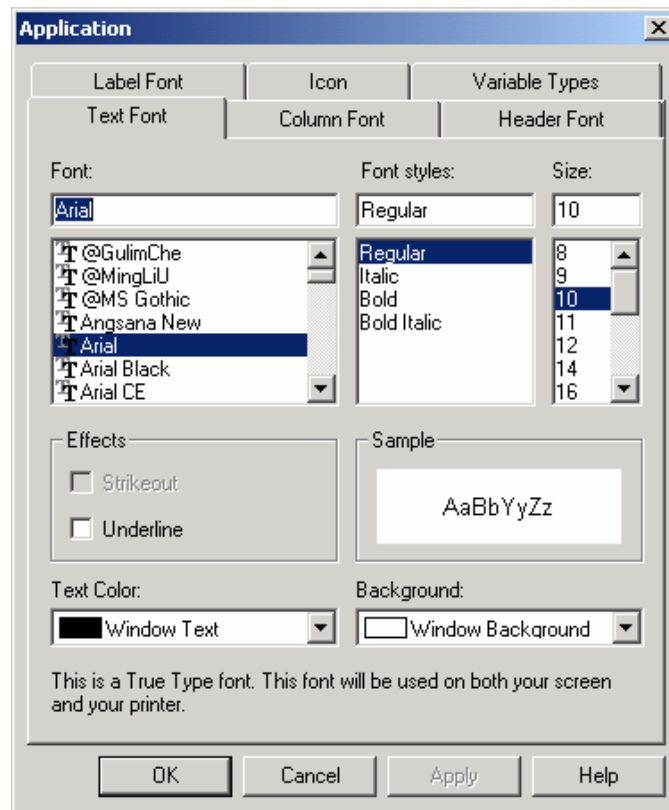
PROPIEDADES DE UNA APLICACIÓN.

Las propiedades de una aplicación son las características globales de la aplicación como por ejemplo las fuentes 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 está.

PROPIEDADES PREDEFINIDAS EN POWER BUILDER

| EVENTO | DESCRIPCION |
|-----------------|--|
| AppName | De sólo lectura, es el nombre del objeto aplicación que se indica al momento de grabar, luego de crearlo. |
| DisplayName | De sólo lectura, es el nombre de la aplicación que es mostrada por otras aplicaciones al utilizar la aplicación. |
| DWMessage Title | Es un título para las cajas de mensaje que se apreciará en la barra de título de las mismas. |

PROPIEDADES ADICIONALES.



Son aquellas como fuente de los textos (Text Font), fuente de las columnas (Column font), Fuente de etiquetas (Label font), fuente de las cabeceras (Header Font), Icono (Icon) y tipos de variables (Variable Types).

Fuente de los textos (Text Font) permite cambiar el tipo de fuente de windows en cuanto a tipo de letra, estilo de la letra (Font styles) como Regular (Normal), Italic (Itálicas) y Bold (negritas) y Bold Italic (en negrita e Italic), tamaño (Size), color de la letra (Text Color), y color de fondo de letra (Background).

EVENTOS DE UNA APLICACIÓN.

Los eventos son todos aquellos sucesos que le ocurren a la aplicación y a los que se puede asignar un script para que se realicen alguna acción cada vez que ocurra el suceso.

EVENTOS PREDEFINIDOS EN POWER BUILDER.

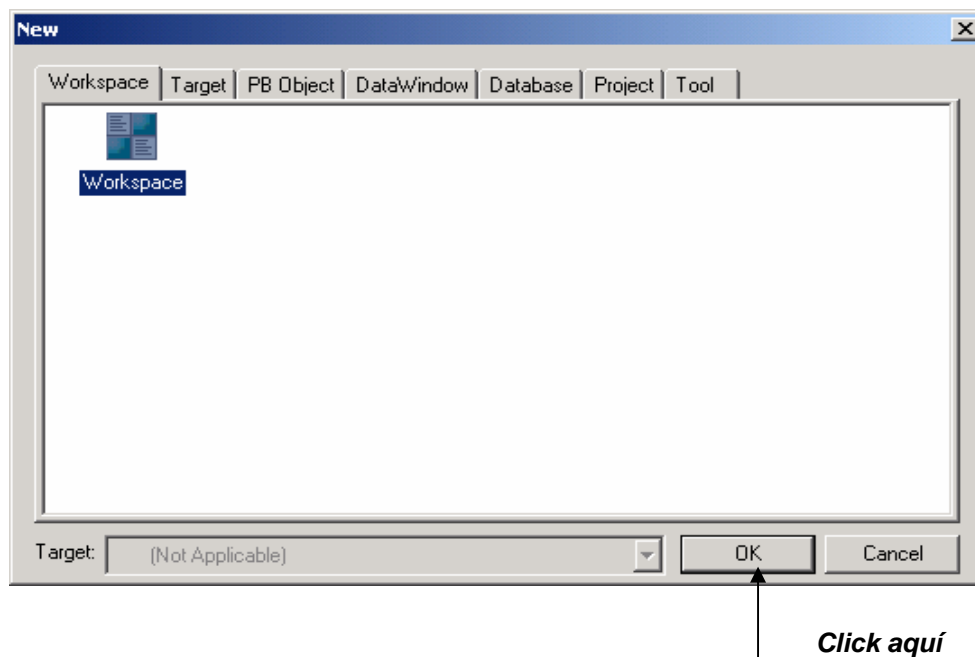
| EVENTO | DESCRIPCIÓN |
|-----------------|--|
| Close | Ocorre cada vez que se termina la ejecución de una aplicación. |
| ConnectionBegin | Obsoleto (usado para conexiones distribuidas con Power Builder) |
| ConnectionEnd | Obsoleto (usado para conexiones distribuidas con Power Builder) |
| Idle | Cuando la función Idle ha sido llamada en un script de una aplicación y el número de segundos que han transcurrido sin que el teclado o el ratón hayan sido usados. |
| Open | Ocorre cada vez que se ejecuta una aplicación. |
| SystemError | Cuando un error crítico ocurre en tiempo de ejecución (como tratar de abrir una aplicación no existente). Si no se pone un script para éste evento Power Builder muestra un mensaje con el número de error y mensaje de error que ha ocurrido. Puede usarse para personalizar los errores. |

A continuación vemos como crear una nueva aplicación.

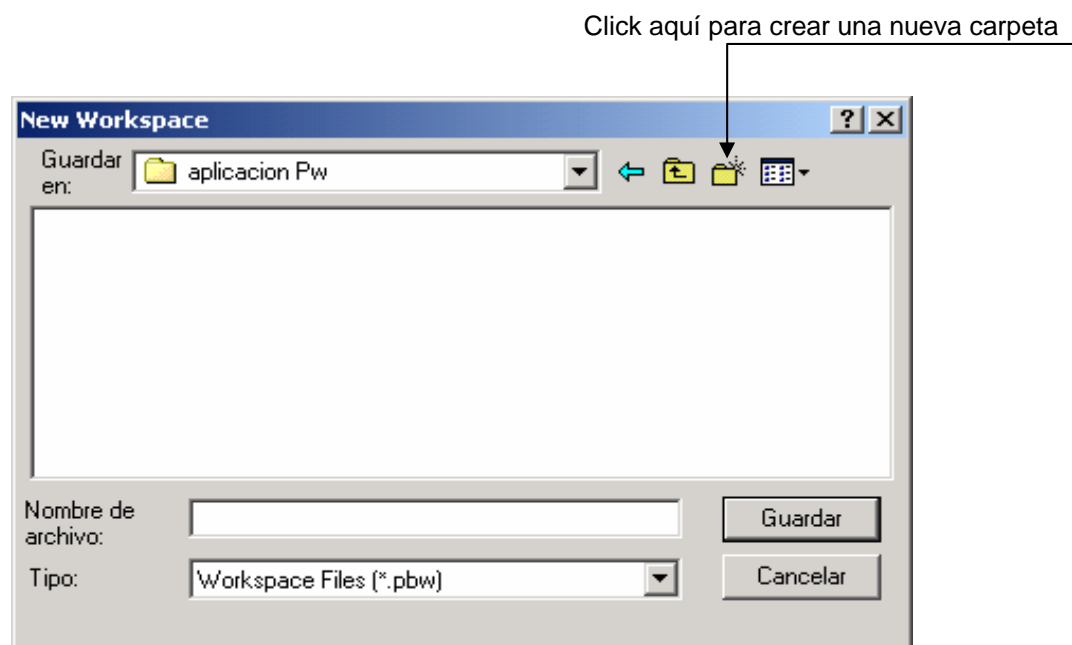
Toda nueva aplicación debe estar contenida en un Workspace(espacio de trabajo), por lo que antes debemos crear el mismo.

Crear un Workspace

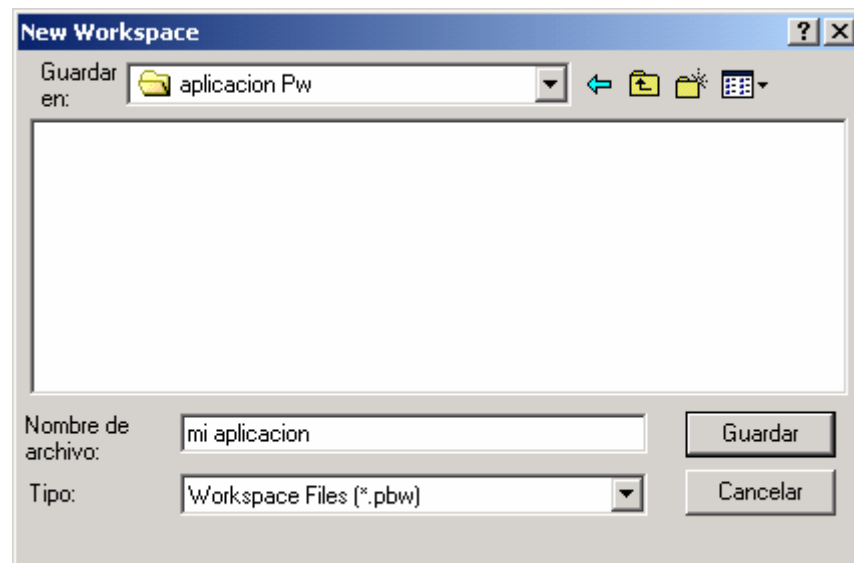
En el menú de Power Builder elija **FILE** y luego **NEW** aparecerá la siguiente ventana:



Después aparecerá :



A la nueva carpeta le daremos de nombre “aplicación PW” (referencia a lo que haremos en nuestra aplicación) y entramos en ella:



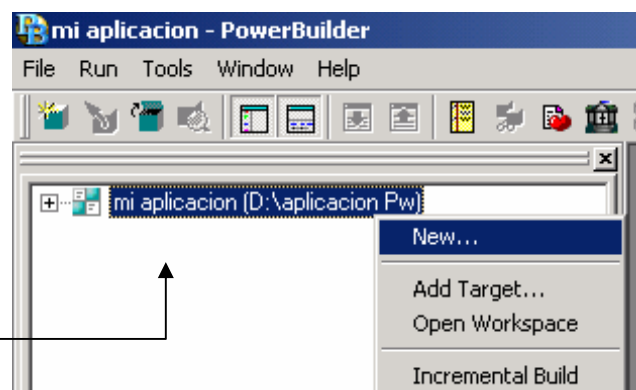
Dentro de dicha carpeta digitamos el nombre del archivo de nuestra aplicación. Ejemplo: “mi aplicación”) y finalmente click en **Guardar**.

Con esto hemos creado el WorkSpace “Mi aplicación”, lo que significa que el archivo WorkSpace se llama: **mi aplicación.pbw**

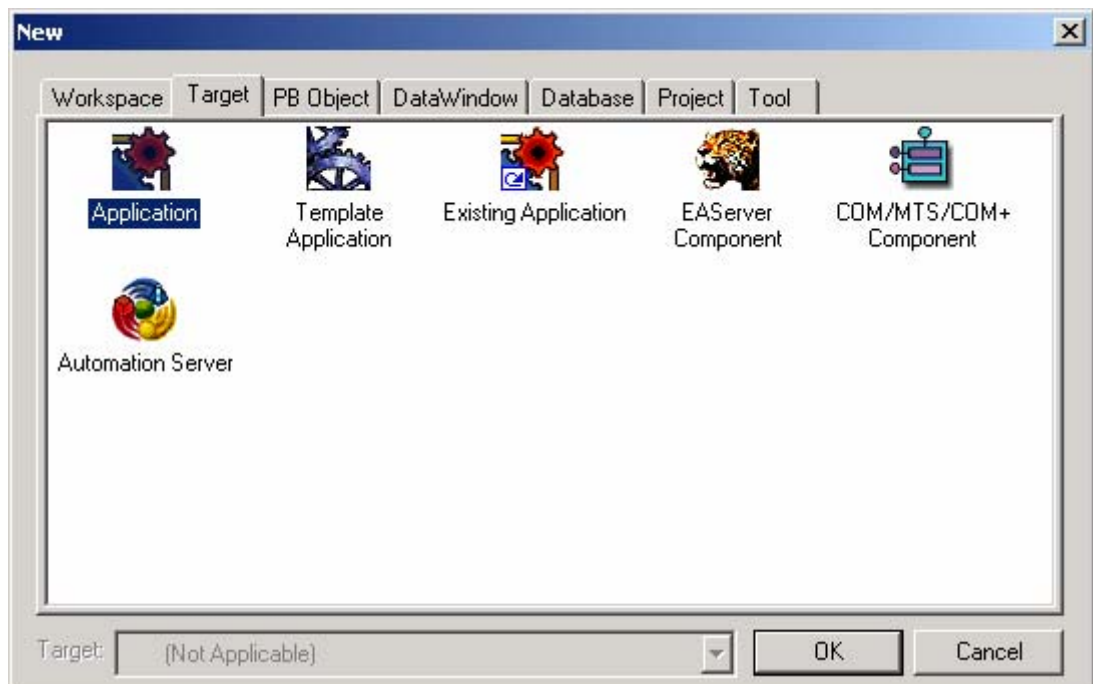
CREAR LA APLICACIÓN.

De Anticlick en el **Workspace Mi aplicación** y elija la opción **New...**

Su nombre de ésta ventana es **System Tree**, en ella aparecerán los objetos que vamos a ir creando.



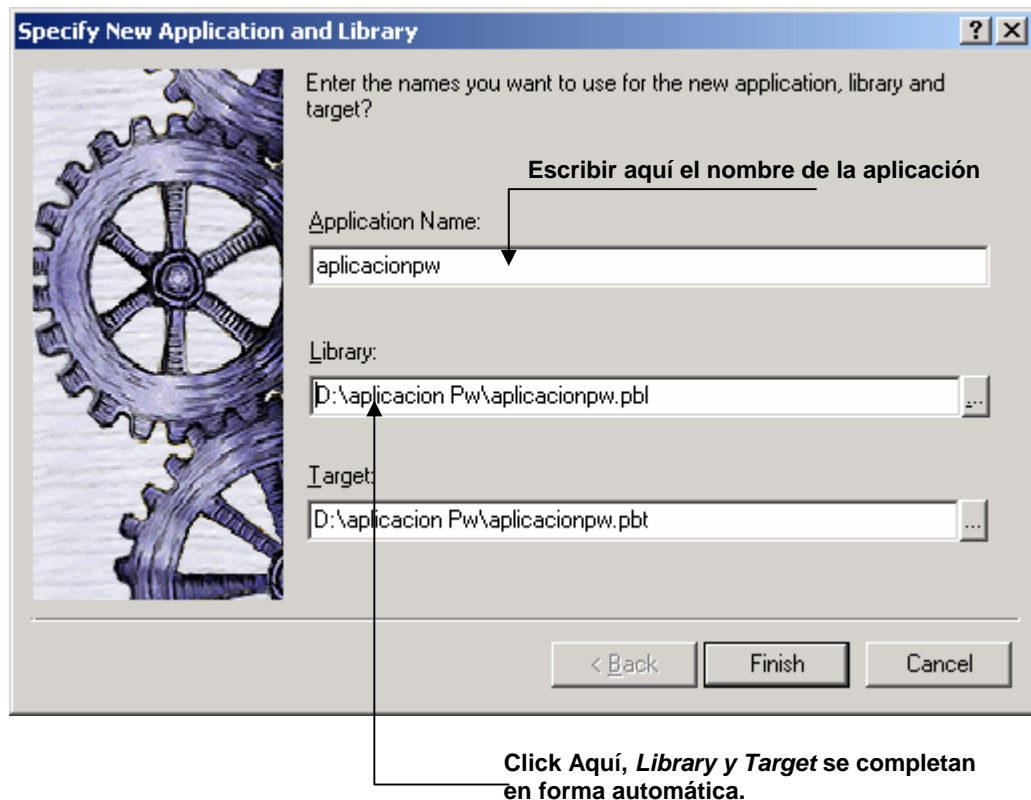
Luego aparecerá la ventana con las siguientes opciones:



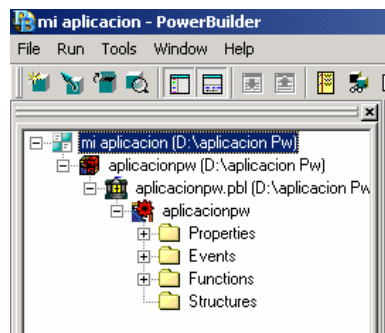
- **Application**: Para una nueva aplicación.
- **Template Application**: Para crear una nueva aplicación que viene con objetos preconstruidos, útil para quienes ya conocen Power Builder.
- **Existing Application**: Para incorporar una aplicación que existe ya sea que esté construida en una versión anterior de Power Builder o no.
- **EAServer Component**: Para crear una aplicación que use componentes de Power Builder usando el Servidor EAServer
- **COM/MTS**: Para crear una aplicación que use componentes Microsoft.
- **Automation Server**: Para crear una aplicación que use una aplicación de Windows como servidor.



De todas ellas elegiremos la primera opción: ***Application***.

Le asignamos un nombre a nuestra primera aplicación. Por ejemplo **aplicacionpw**.

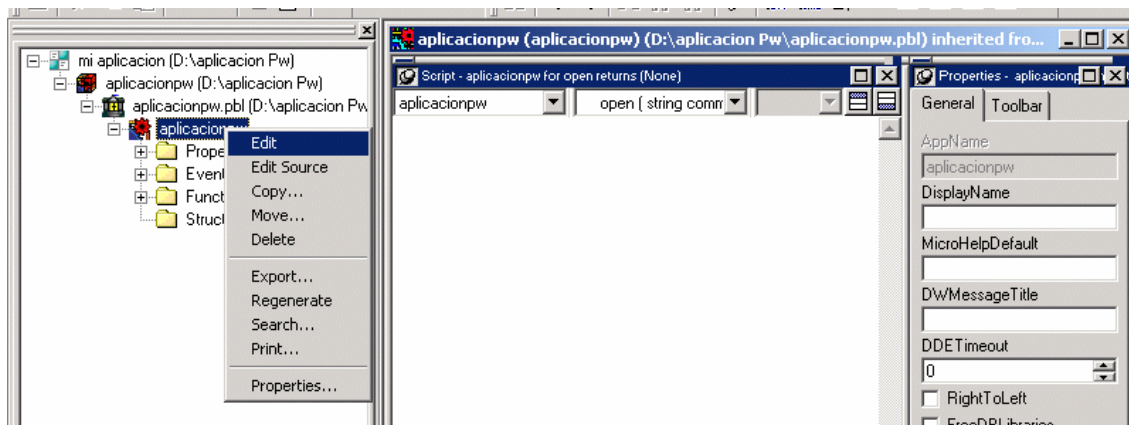


Despliegue el árbol del sistema (System Tree) y se muestra de la siguiente manera



Donde observes este símbolo  da click en él para desplegar una rama del árbol. Cuando una rama está desplegada le antecede el siguiente símbolo 

Realizamos el despliegado hasta ver lo mismo que en la figura. Para entrar al código y propiedades de esta nueva aplicación da Anticlick en la aplicación **aplicacionpw** y a continuación click en **Edit**.

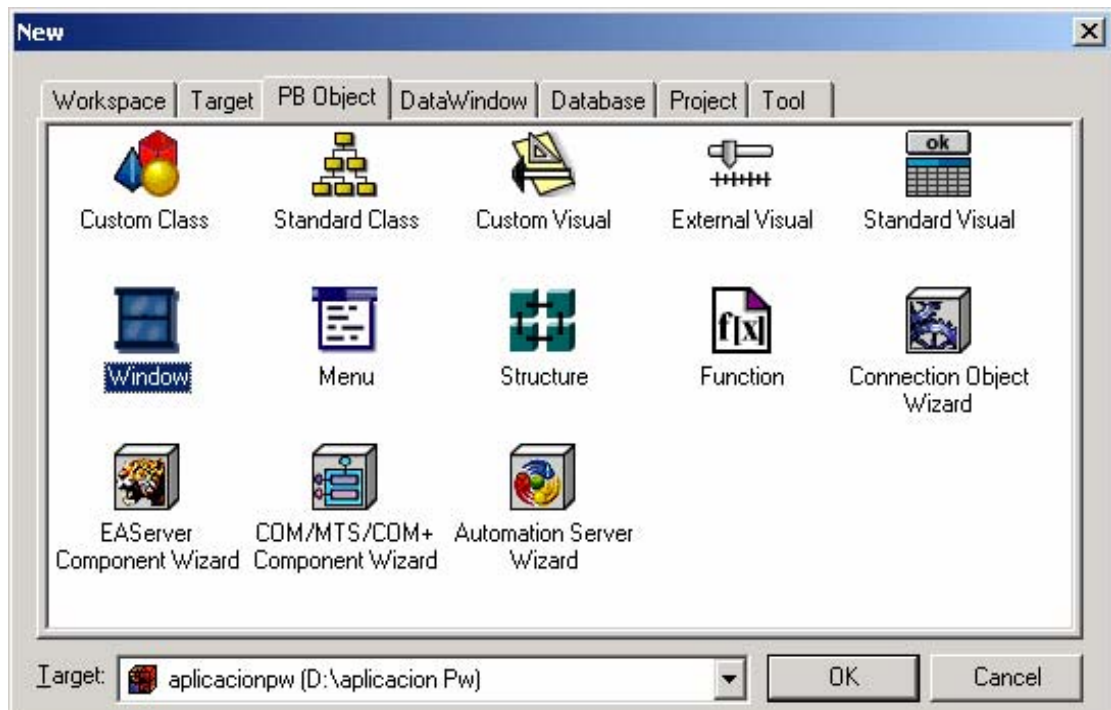


Se muestran dos ventanas a la derecha del System Tree (árbol del sistema)

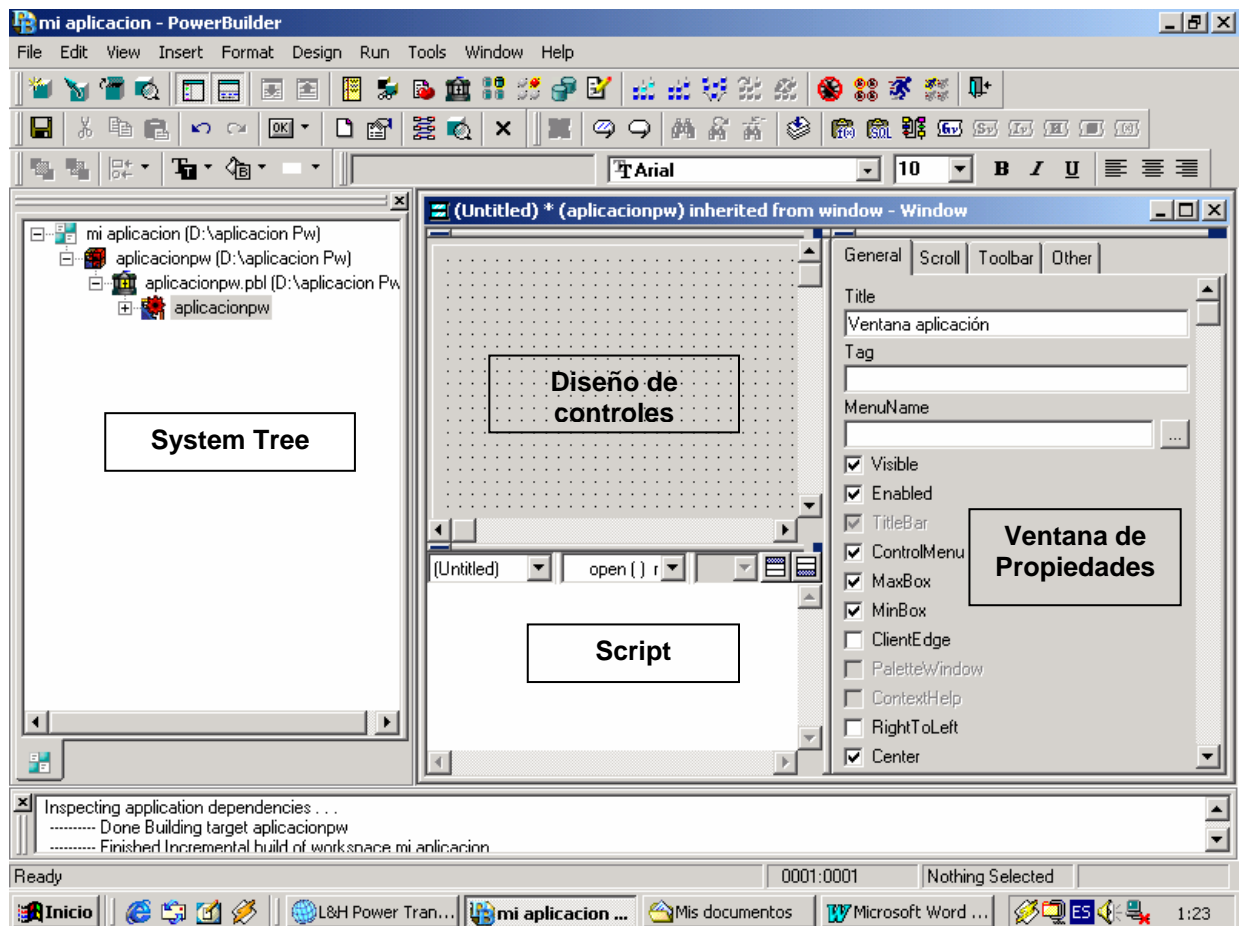
CAPITULO V

CREACION DE WINDOW (VENTANA)

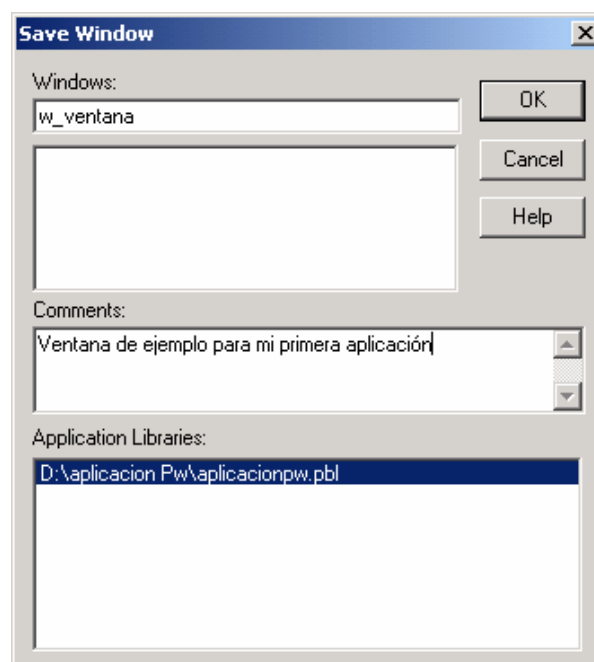
Para crear la primera ventana seleccione **FILE** seguidamente **NEW...** (o CTRL + N) en la **ficha PB Object** seleccionar el objeto **Window**.



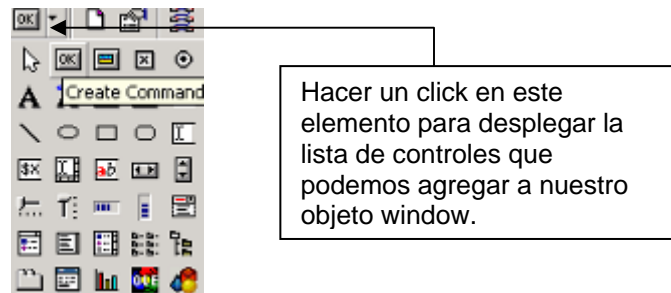
Se muestra un área de diseño de ventanas (formulario), seleccionaremos la propiedad **Title** y digitaremos el título de la ventana. Ejemplo: “Ventana de aplicación”. Si la Ventana de **Properties** no aparece dar anticlick en el **Layout** y elegir **Properties** para que aparezca.



Luego grabaremos la ventana con el icono de Guardar o eligiendo **FILE** y después **SAVE** en el Power Menú.



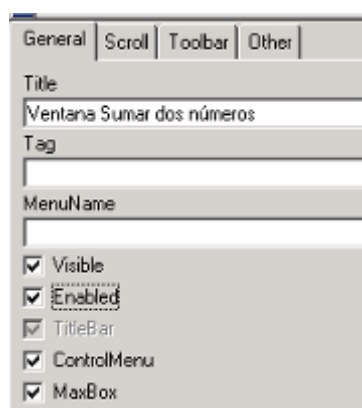
A partir de este momento agregar los controles que sean necesarios para nuestra primera aplicación, así por ejemplo:



Esta ventana principal tiene 5 zonas esenciales para poder iniciar a diseñar nuestra aplicación

1. **System Tree** (Arbol de Sistema). Muestra todos los objetos visibles o no visibles que vamos creando para nuestra aplicación.
2. **Ventana de diseño de controles**. Donde pegaremos todos los controles que requiera nuestra aplicación, el primer objeto que vemos allí es el window.
3. **La ventana de codificación (Script)**. En esta ventana se codifica todos los objetos según el evento que deseamos que se desencadene o suceda.
4. **La ventana de Propiedades**. En esta ventana encontramos todas las características o atributos referentes al control seleccionado.
5. **Ventana de Salida (Output)**. En esta ventana nos mostrará la salida de los sucesos ocurridos durante la compilación de la aplicación.

Comenzaremos nuestra aplicación colocando el nombre que llevará nuestra ventana cuando se ejecute para ello vamos a la ventana de propiedades y en la opción **Title**, escribimos “Ventana Sumar dos números”.



Además de la ficha **General** también vemos 3 opciones más :

PROPIEDADES DE VENTANA

Las propiedades de la ventana están organizadas en forma de FICHAS, aquí la descripción de sus principales características:

| FICHA | DESCRIPCION |
|---------|--|
| General | <p>La casilla TITLE, indica el texto de título 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">• Child!, ventana que permanece inserta dentro de una ventana MDI• Main!, ventana normal (principal) con posibilidad de menú.• Mdi!, ventana que posee área interna para contener otras ventanas.• MdiHelp!, igual que la anterior, y sirve como ayuda.• Popup!, ventana de contexto o emergente.• Response!, necesita cerrar esta ventana para continuar la ejecución. |
| Scroll | Permite definir si la ventana tendrá barras de desplazamiento, y la unidad a utilizarse por ellas. |
| Toolbar | Permite definir si se utilizará la caja de herramientas definida para el menú su presentación y posición. |
| Other | Permite indicar la posición y tamaño de la ventana dentro de la pantalla, como será presentada. |

CAPITULO VI

CONTROLES

En el desarrollo de software se ha intentado introducir elementos propios de la ingeniería a fin de aprovechar el esfuerzo que supone la construcción de sistemas informáticos. Una de las áreas en las que mas se ha avanzado en este sentido es la que intenta convertir el proceso de construcción de programas es un proceso de conjunción de elementos ya existentes.

En la construcción de un automóvil se realizan labores creativas, como por ejemplo, el diseño de la carrocería, pero también una labor de ensamblaje de elementos: el motor, las ruedas y los distintos componentes, normalmente ya existentes antes de la creación del nuevo modelo. Lo único que se hace es unir estos elementos.

Los controles de ventanas de Power Builder son objetos visuales preconstruidos que se ubican en cualquier zona de las ventanas y permiten armar una interfaz de usuario completa combinándolos a criterio del diseñador de sistemas.



En el painterBar de **Window** existe el botón que permite elegir a dichos controles y es éste:






















Muestra el listado de todos los controles que es posible de utilizar . Adicionalmente se pueden diseñar objetos con características adicionales para ser utilizadas también.






Para añadir un control a la ventana, seleccione el control, del listado. Haga click sobre el lugar en la ventana donde desea añadirlo. Para seleccionar un control de la ventana, de click en él, o mantenga presionada la tecla control para seleccionar varios objetos. Puede moverlos arrastrando el mouse.

Los controles son los siguientes :

| Control | Icono | Prefijo | Descripción |
|---------------|---|---------|---|
| CommandButton |  | cb_ | Para implementar botones, con los que se inician procesos o por el contrario para terminarlos, cancelarlos o cerrar ventanas. |
| PictureButton |  | pb_ | Botón con imágenes. Tiene casi todas las características del control anterior, pero |

| | | | |
|------------------|---|------|--|
| | | | adicionalmente permite mostrar una figura dentro de él. |
| CheckBox |  | cbx_ | Permite hacer una selección (marcarlo) entre varias alternativas. Cada alternativa es independiente. |
| RadioButton |  | rb_ | Permite hacer una selección (marcarlo) entre varias alternativas. Sólo una alternativa, del conjunto puede seleccionarse. |
| StaticText |  | st_ | Caja de texto, como etiqueta. Solo para mostrar información, no hay posibilidad que pueda escribirse directamente en ella. |
| StaticHyperLink |  | shl_ | Muestra información y además permite definir una URL para cargar la página web que se le indique. |
| Picture |  | p_ | Caja de dibujo. Permite visualizar una figura. Esta puede ser definida en tiempo de diseño, puede ser modificada en tiempo de ejecución. |
| PictureHyperLink |  | phl_ | Muestra gráficos y además permite definir una URL para cargar la página web que se le indique. |
| GroupBox |  | gb_ | Se usa para agrupar cualquier grupo de controles, por ejemplo es muy útil para agrupar los controles RadioButton y CheckBox. |
| Line |  | l_ | Permite dibujar una línea. |
| Oval |  | ov_ | Permite dibujar un óvalo. |
| Rectangle |  | r_ | Permite dibujar un rectángulo. |
| RoundRectangle |  | rr_ | Permite dibujar un rectángulo con bordes redondeados. |
| SingleLineEdit |  | sle_ | Caja de edición de texto, es posible que el usuario en tiempo de ejecución escriba un texto y se vean sólo |

| | | | |
|------------------------|---|--------|---|
| | | | asteriscos, se usa para el ingreso de claves de acceso (passwords). |
| EditMask |  | em_ | Caja de texto, a la que se le puede restringir el conjunto de caracteres a ingresar. Permite un formato de presentación. |
| MultiLineEdit |  | mle_ | Se usa para ingresar varias líneas en una entrada de datos. |
| RichTextEdit |  | rte_ | Se usa para mostrar texto enriquecido, es decir texto coloreado, en negritas, itálicas, subrayado, etc. |
| HScrollBar |  | hsb_ | Barra de desplazamiento horizontal. |
| VScrollBar |  | vsb_ | Barra de desplazamiento vertical. |
| HtrackBar |  | htb_ | Barra de pistas horizontal. |
| VtrackBar |  | vtb_ | Barra de pistas vertical. |
| HprogressBar |  | hsb_ | Barra de progreso horizontal. |
| VprogressBar |  | vsb_ | Barra de progreso vertical. |
| DropDownListBox |  | ddlb_ | Caja de lista desplegable. Muestra un solo ítem, el seleccionado, de la lista que se despliega al presionar el botón adjunto. |
| DropDownPictureListBox |  | ddplb_ | Similar al anterior pero soporta figuras por cada elemento que muestra. |
| PictureListBox |  | plb_ | Es una caja de lista con elementos gráficos y de texto. |
| ListBox |  | lb_ | Caja de lista. Muestra todos los ítems a la vez, pudiendo seleccionarse inclusive, más de uno. |
| ListView |  | lv_ | Se usa para crear una vista de objetos. |
| TreeView |  | tv_ | Se usa para crear una vista de tipo árbol, con una raíz común y varios nodos y elementos colgados de dichos nodos. |

| | | | |
|------------|---|------|---|
| Tab |  | tab_ | Es un separador de páginas que son contenedoras de objetos visuales de Power Builder. |
| DataWindow |  | dw_ | Datawindow Control, el que se vincula con un objeto Datawindow diseñado. |
| Graph |  | g_ | Se usa para crear gráficos estadísticos. |
| Ole |  | ole_ | Se usa para incorporar objetos de windows de otras aplicaciones distintas al Power Builder. |
| USerObjetc |  | uo_ | Se usa para incorporar objetos de usuario creados previamente por alguien usando objetos visuales y objetos no visuales de Power Builder preconstruido. |

PROPIEDADES DE LOS CONTROLES

Las propiedades de los controles son las características y pueden ser modificadas tanto en tiempo de diseño, como en tiempo de ejecución; mediante el uso de sentencias. Para ver las propiedades de un control, de anticlick en él y elija la opción **Properties** del menú que se despliega.

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

| FICHA | DESCRIPCION |
|---------|---|
| General | <p>“Name” indica el nombre del objeto.</p> <p>La casilla “Text”, indica el texto a ser mostrado.</p> <p>“Tag”, permite guardar cualquier dato.</p> <p>“Visible”, indica si el objeto puede verse.</p> <p>“Enabled”, indica si el objeto esta activamente disponible.</p> <p><u>Para los botones:</u> “Default”, indica que si se presiona el Enter en cualquier parte de la ventana, se activa dicho botón.</p> <p>“Cancel”, indica que si se presiona la tecla Escape, se activa dicho botón.</p> <p><u>Para las cajas de Edición, Etiqueta y Lista Desplegable:</u></p> <p>“Alignment”, indica la alineación del texto,</p> <p>“Case”, el tipo de presentación.</p> <p>“Border”, indica el tipo de borde.</p> <p>“Limit”, el número de caracteres a dejar ingresar.</p> |

| | |
|----------|---|
| | <p><u>Para la lista desplegable.</u> "Allow Edit", indica que se puede escribir un ítem además de seleccionarlo.</p> <p>"Sort", indica que el listado será ordenado alfabéticamente.</p> <p><u>Para la Lista:</u> "Multi Select", indica posibilidad de selección múltiple.</p> |
| Font | <p>Permite definir el tipo de letra, tamaño y características de presentación del texto visible. "Text Color", indica el color del texto.</p> <p>"Background", indica el color de fondo.</p> |
| Position | Indica las coordenadas "X" e "Y", también se indica el ancho (Width) y el alto del objeto (Height) |
| DragIcon | Indica el tipo de icono, que será utilizado por el cursor cuando este pase por encima del objeto en tiempo de ejecución. |
| Items | Para los objetos lista, permite ingresar el texto de cada ítem. |
| Mask | <p>Para la máscara, permite definir la máscara de entrada de datos. Seleccione "MaskDataType" para ver en "Masks", los posibles caracteres a utilizar. Cualquier otro carácter se considera un literal.</p> |

CAPITULO VII

PROGRAMACION ORIENTADA A EVENTOS

CONFECCIÓN DE UN PLAN DE EVENTOS

El plan de eventos es una herramienta efectiva que nos permite planificar el comportamiento de nuestra interfaz gráfica ante los posibles eventos que puede desencadenar el usuario, determinando que eventos de todos los posibles se van a controlar y en que objetos se va a controlar la ejecución de los mismos.

Además podemos listar todas las actividades que se llevaran a cabo cuando a un determinado objeto le ocurra un evento que deseamos manejar.

Por lo que el plan de eventos considerará en una primera fila el nombre de la ventana(window) a continuación en una primera columna todos los objetos para los que deseamos controlar eventos, en la segunda columna los nombres de los eventos que controlaremos para cada control y en la tercera columna las actividades que se llevarán a cabo cuando a un objeto le ocurra un evento.

Plan de Eventos para w_ventana

| Nombre de la Ventana : | | |
|------------------------|-----------|---|
| Controles | Eventos | Actividades |
| sle_nro1 | Losefocus | Verificar que se haya ingresado un valor correcto y desplazar el cursor a la siguiente caja de texto. |
| sle_nro2 | Losefocus | Verificar que se haya ingresado un valor correcto y desplazar el cursor al botón correspondiente. |
| cb_sumar | Clicked | Calcular la suma entre los datos ingresados en las cajas de texto: nro1 y nro2. |
| Cb_cancelar | Clicked | Limpiar las cajas de texto y colocar el cursor en la caja de texto nro1. |
| Cb_salir | Clicked | Cerrar la ventana |

PROGRAMACIÓN DE LOS EVENTOS (SCRIPTS)

Para agregar un comentario en los scripts utilizar los símbolos "//" (dos barras diagonales)

Para el objeto **sle_nro1**:

| Sle_nro1 | Losefocus |
|--|-----------|
| // Verificar que se haya ingresado un valor correcto y desplazar el enfoque a la siguiente caja de texto | |
| integer n1 | |
| n1=integer(sle_nro1.text) | |
| if n1>0 then | |
| sle_nro2.setfocus() | |
| else | |
| sle_nro1.text="" | |
| sle_nro1.setfocus() | |
| end if | |

Para el objeto **sle_nro2**:

| Sle_nro2 | Losefocus |
|--|-----------|
| // Verificar que se haya ingresado un valor correcto y desplazar el enfoque al botón correspondiente | |
| integer n2 | |
| n2=integer(sle_nro2.text) | |
| if n2>0 then | |
| cb_sumar.setfocus() | |
| else | |
| sle_nro2.text="" | |
| sle_nro2.setfocus() | |
| end if | |

Para el objeto **cb_sumar**:

| | |
|---|---------|
| Cb_sumar | Clicked |
| <pre>// Calcular la suma entre los datos ingresados en las cajas de texto: nro1 y nro2. integer suma suma=(n1+n2) //se realiza la suma messagebox("hola",string(suma))</pre> | |

Para el objeto **cb_cancelar**:

| | |
|---|---------|
| Cb_cancelar | Clicked |
| <pre>// Limpiar las cajas de texto y colocar el cursor en la caja de texto nro1. sle_nro1.text="" sle_nro2.text="" sle_nro1.setfocus()</pre> | |

Para el objeto **cb_salir**:


| | |
|--|---------|
| Cb_salir | Clicked |
| <pre>// Cerrar la ventana. Close(parent) //cerrará el objeto padre del botón, en éste caso la ventana que lo contiene</pre> | |

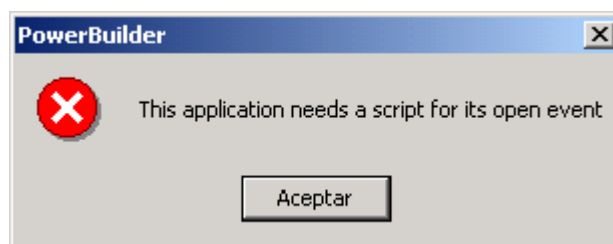
Declaración de variables:

| | |
|--|-------------------|
| Declare | Instance Variable |
| <pre>//declarar las variables a utilizar de tipo entero integer n1,n2</pre> | |


CAPITULO VIII

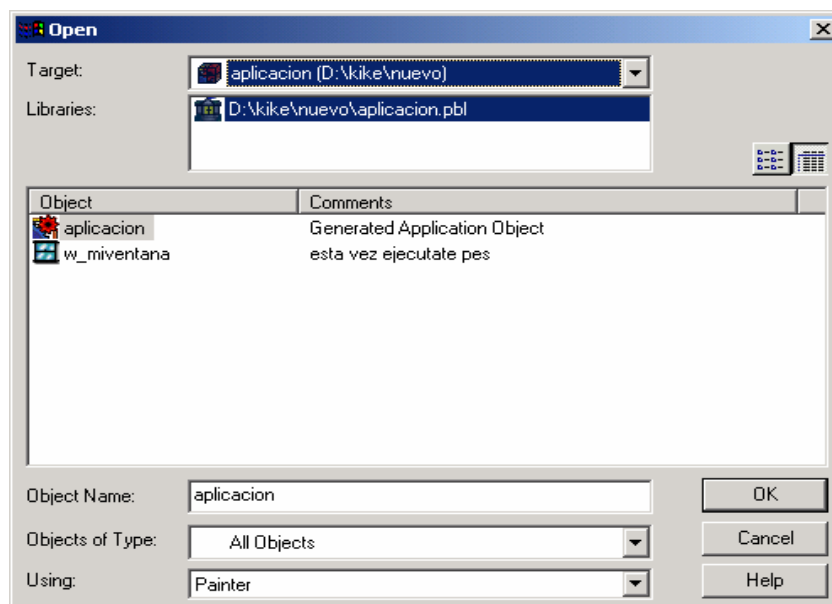
ESCRIBIR CODIGO, COMPILAR Y EJECUTAR LA APLICACIÓN

Existe una última tarea a realizar antes de que pueda correr la aplicación y llevarlo a cabo. Debe escribir un script que diga que la aplicación “aplicacionpw” se abra y muestre la ventana “w_ventana” ejecutándose, si trata de abrir la aplicación ahora mismo haciendo un click al botón RUN  del PowerBar, recibirá un mensaje de error “esta aplicación necesita un script para su evento open”



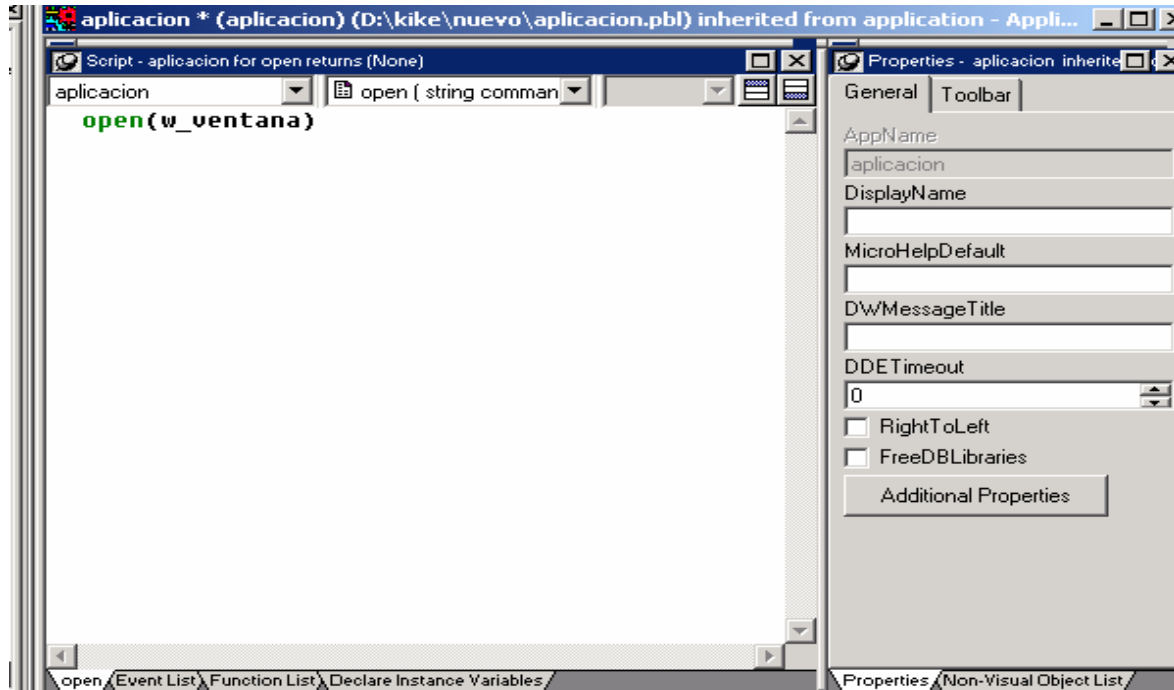
Para evitar este problema lo que tenemos que hacer es realizar uno de lo siguientes procedimientos:

1. Seleccionar del PowerBar1 el ícono  se presentará la ventana de diálogo **OPEN**, del control DropDownListBox Objects of types: Seleccionar **Applications** a continuación seleccionar el nombre de su aplicación, tal como se muestra en la figura siguiente:
2. En la ventana System Tree hacer un anticlick en el objeto **aplicación** luego click en **Edit** :
3. Click en Menú **FILE** luego click en **OPEN**





Después de dar un click en el botón OK se presentará una ventana de edición (Script) en la cual se debe escribir el código correspondiente al evento seleccionado.

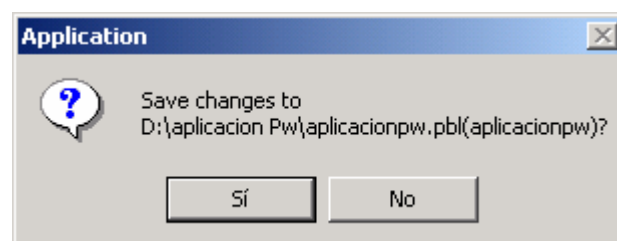
Open(w_ventana)



COMPILACION DEL SCRIPT

Haga un click en el icono compile  en el painterBar1 o presionando CTRL + L para compilar el Script, si tiene errores de sintaxis hay que corregirlos para que no se muestren los mensajes de error.

Finalmente ya estamos listos para visualizar y ejecutar nuestra aplicación, para ello hacer Click en el botón RUN  del PowerBar1. Power Builder preguntará si desea grabar los cambios en la aplicación?



Hacemos click en el botón SI y muestra la aplicación que hemos desarrollado poniendo punto final a nuestra PRIMERA APLICACIÓN.