

# CAPITULO VI

## FASES DE LA IMPLEMENTACIÓN DEL APLICATIVO

Para la realización del aplicativo fue necesario una minuciosa investigación sobre el manejo y configuración de las herramientas de software que se utilizaron durante el desarrollo. Entre las que se puede anotar:

- Power Designer 9.0
- Microsoft SQL Server 2000
- Power Builder 8.0
- Microsoft Office (Excel y Word)

También se revisó sobre asuntos de hardware, con el fin de conocer las características de la arquitectura cliente-servidor, es decir las especificaciones apropiadas para tener un buen rendimiento durante una conexión en red.

Enseguida se detalla todas las etapas que se ejecutaron para implementar el sistema de crudos.

### **6.1 Investigación preliminar**

Se comenzó investigando todo tipo de información referente al manejo del lenguaje de programación PowerBuilder, la base de datos SQL Server y otras herramientas que fueron indispensables para el diseño y desarrollo óptimo del sistema.

Diferentes fuentes de información se ha utilizado para el diseño y desarrollo del aplicativo mencionado en la presente tesis, entre las más importantes están:

- Libros y manuales
- Revistas y Folletos
- Internet
- Monografías
- Ayuda On-line /Off-line

## 6.2 Análisis de la información

Toda información recopilada, fue analizada y documentada de forma resumida y concisa.

El estudio de las bondades del lenguaje de programación Power Builder es muy extenso, por esta razón ha sido necesario examinar desde su entorno de programación hasta las relaciones de conexión que tiene con otras herramientas a través de una lista ya establecida, que se muestra a continuación:

- Configuraciones previas
- Entorno de programación
- Codificación de funciones, procedimientos, eventos, etc.
- Objetos y/o Controles
  - o Workspace, Datawindow, Datastore, Window, Menu, UserObject, Application, function, standard visual, Database, entre otros.
- Módulos
- Conectividad con Base de datos

Una parte muy importante del software a utilizar es el tipo de instalación que se debe escoger, debido a que Power Builder viene en tres versiones como se lo indica en el Capítulo IV. Existen paquetes o herramientas adicionales que vienen con este lenguaje de programación, las mismas que ayudan en gran medida para el desarrollo de aplicaciones en general, algunas de estas son:

**Native Database Interfaces.-** Controladores nativos de Base de datos para el servidor.

**ODBC Database Drivers.-** Interface para conectar a la mayoría de ODBC.

**OLE DB Interface.-** Interface para conectar a las fuentes de datos OLE DB

**JDBC DRIVERS.-** Controladores para conectar a fuentes de datos JDBC.

**Foundation Class Library(PFC).-** Conjunto de clases reusables, incluye ejemplos en código y aplicaciones.

**Demo Database.-** Base de datos de demostración de PowerSoft, utilizada para ejemplos de código.

**Translation Toolkit.-** Herramientas para la traducción de aplicaciones.

**Web Plugins.-** Colección de objetos para activar en aplicaciones para Internet.

**SCC Interface.-** Controlador genérico para la conexión a cualquier fuente que soporte la interface SCC para Microsoft.

**Code Examples.-** Aplicaciones con código que muestran las principales técnicas de codificación.

Estas herramientas las podemos seleccionar desde el tipo de instalación personalizada (Custom).

### **Conexión a la base de datos**

En la programación muchas veces los problemas surgen cuando se debe configurar una conexión a la base de datos. Cuando la solución está en que al momento de instalar la herramienta de programación debemos tomar en cuenta los drivers (ODBC) para el acceso a los datos desde el cliente. PowerBuilder ofrece la instalación de drivers ODBC compatibles para la mayoría de base de datos del mercado.

SQL Server permite el acceso a los datos por medio del estándar ODBC, por lo que debemos seleccionar el del SQL Server.

Durante el desarrollo de una aplicación suele suceder muchas veces que se debe programar desde un equipo cliente y no precisamente en el equipo servidor, lo que hace necesario instalar la herramienta de programación en un ambiente cliente-servidor. En la siguiente sección se menciona esta situación.

### Instalación del Lenguaje de programación en Red

Una gran verdad es que las actualizaciones oficiales de PowerBuilder se dan cada seis meses aproximadamente, pero también la publicación de los conocidos bugs o parches se presentan más seguido por medio de la internet. Por tanto, si se tiene varios puestos de desarrollo de aplicaciones, sería muy demoroso actualizar equipo por equipo según las instalaciones locales que se hayan realizado.

Una solución más fiable es hacer uso de una red de área local, donde se realice una única instalación del software de programación y que a la vez permita ser accesible por todos los equipos de desarrollo. La siguiente figura muestra esta arquitectura.

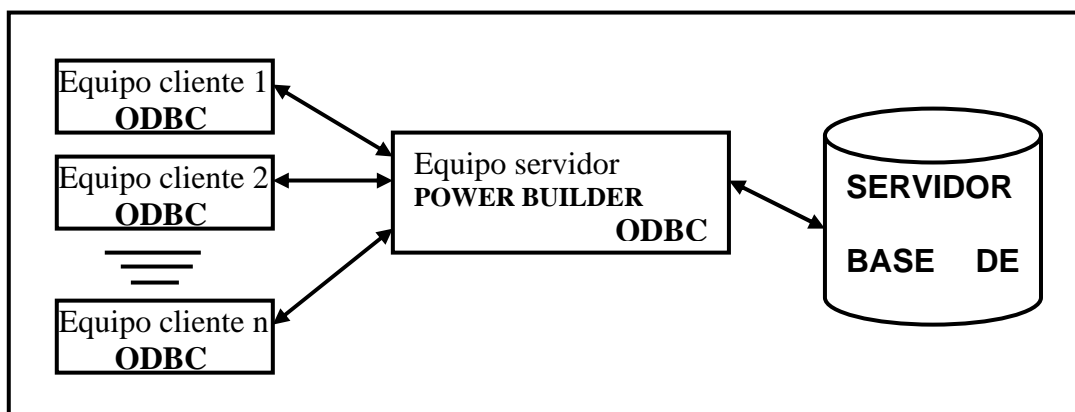


Figura 6.1. Arquitectura cliente –servidor

Hay dos formas de instalar el software: desde el servidor o desde un equipo cliente indicando la carpeta disponible en el servidor. Pero se

puede observar que surgen nuevos problemas en este tipo de instalación, tales como:

- **No se puede ejecutar el software desde el cliente**

Se debe a que algunas librerías DLL no están disponibles o no se puede tener acceso directo. Lo correcto es digitar en el archivo autoexec.bat en la variable PATH la dirección de los archivos comunes:

```
C:\Program Files\common files\shared
```

Otra solución es usar el registro del sistema, con el fin de indicar la ruta del ejecutable. El comando regedt32.exe nos permite esta acción. HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\AppPaths

Aquí visualizamos la lista de programas ejecutables del sistema operativo, entonces debemos definir una aplicación nueva con el nombre PB60, para esto realizamos los siguientes pasos:

- Menú Edición → Nuevo → Clase
- y digitar PB60.EXE
- y como clase digitar REG\_EXPAND\_SZ
- Luego ingresar dos valores, el directorio donde se está instalado Power Builder en red y el directorio de las DLL que utiliza este software; esto se realiza mediante el menú Edición y Agregar valores y para las DLL en el nombre de la clase digitamos Path y en el siguiente cuadro de texto indicamos la ruta de estas DLL.

- **No se abre la ayuda del software**

Este problema es del mismo tipo del anterior, donde el equipo cliente no puede hallar las DLL necesarias. Su solución también se basa en la edición del archivo del sistema autoexec.bat o por medio del comando de registro de sistema. En este segundo caso se debe ubicar la parte donde se registra la ruta de las Ayudas: HKEY\_LOCAL\_MACHINE\Software\PowerSoft\PowerSoftHelp\InipathsForHelp. Aquí digitamos el valor PB60 con la clase REG\_SZ junto con el valor de la cadena que es la ruta de red donde están los ficheros HLP.

- **No se realiza la conexión a la base de datos**

Muchas veces se debe este problema a que estamos trabajando con otro tipo de base de datos que no son de la familia de Sybase, en este caso se utiliza la Base de datos SQL Server, pero no hay ningún problema si se utiliza el ODBC correcto. Este problema también puede ser porque no se ha compartido los directorios de la base de datos.

C:\sqlany50 y c:\sqlany50\win32.

### **Aspectos en la configuración de la Base de Datos SQL Server**

Por otra parte, el estudio de la Base de datos SQL Server y sus herramientas de cliente-servidor también se han resumido en la siguiente lista:

- Configuración para cliente y servidor
- Seguridades
- Codificación en el entorno del Analizador de Consultas
- Manejo del Administrador Corporativo
- Creación de base de datos, vistas, procedimientos almacenados y disparadores.
- Creación y configuración de usuarios y privilegios
- Conectividad ODBC y perfiles

### 6.3 Análisis y Estudio del entorno de programación

El entorno de programación es muy amigable y por tanto cualquier usuario conocedor de las distintas herramientas de programación puede familiarizarse muy rápidamente.

#### Entorno de desarrollo del lenguaje de programación

PowerBuilder posee un entorno de programación completo, ya que tiene todas las funciones integradas para la solución de aplicaciones complejas.

Cada espacio de trabajo de un elemento de programación, se le conoce como Painter. Ejemplos de algunos painter son los siguientes:

Aplicación.- Creación de clases tipo Aplicación

Ventana.- Creación de clases de interfaz

Menú.- Creación de Menús de la aplicación

Datawindow.- Creación de Clases para acceso a datos externos

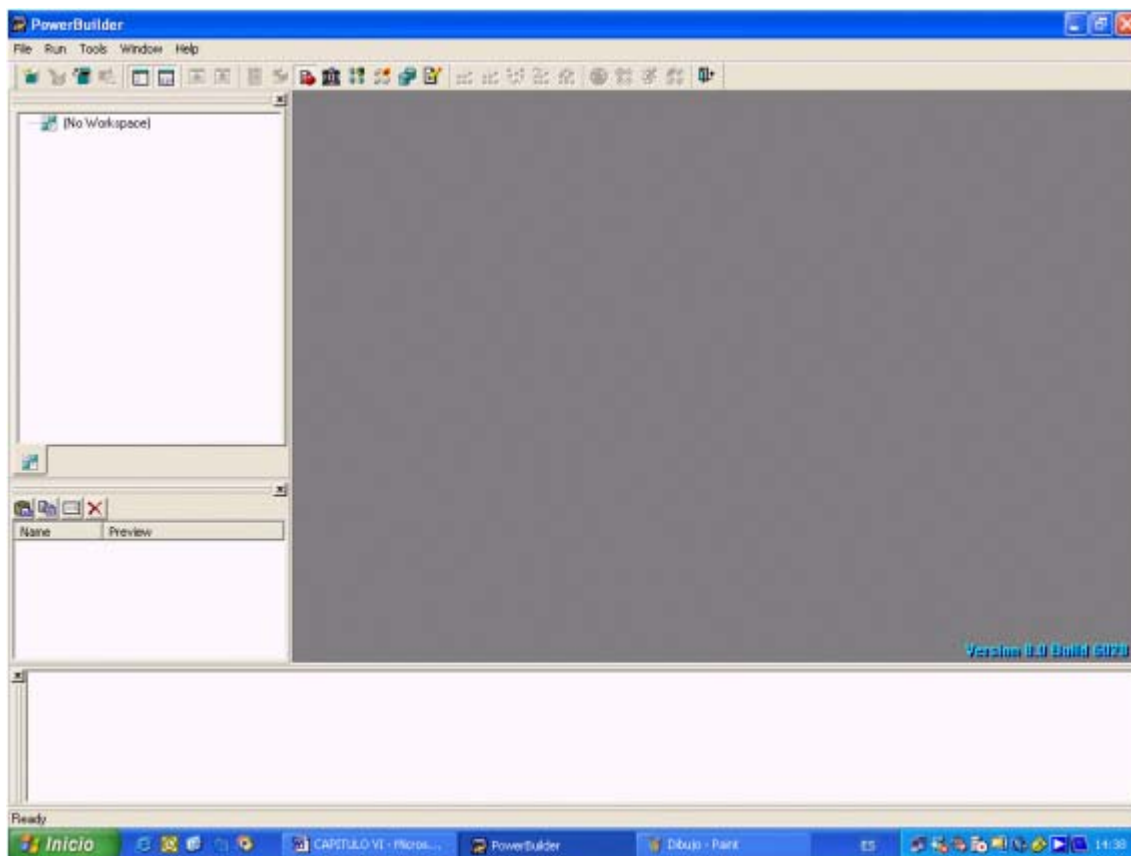
Edición.- Editor de texto

Ejecutar Ventana.- Abre una ventana

Salir.- Finaliza el entorno.

Estos painter mencionados se muestran al iniciar una aplicación en PowerBuilder, y pueden ser ejecutados según una jerarquía establecida.





**Figura 6.2** : Entorno de programación de PowerBuilder 8.0

La versión en la que se realizó esta aplicación es en la 8.0 y la jerarquía de los elementos para su programación es:

1. Creación del Painter Workspace
2. Application
3. Cualquier objeto que va ha ser parte de la aplicación (window, menu, datawindow, etc.)

### **Elementos que conforman una aplicación**

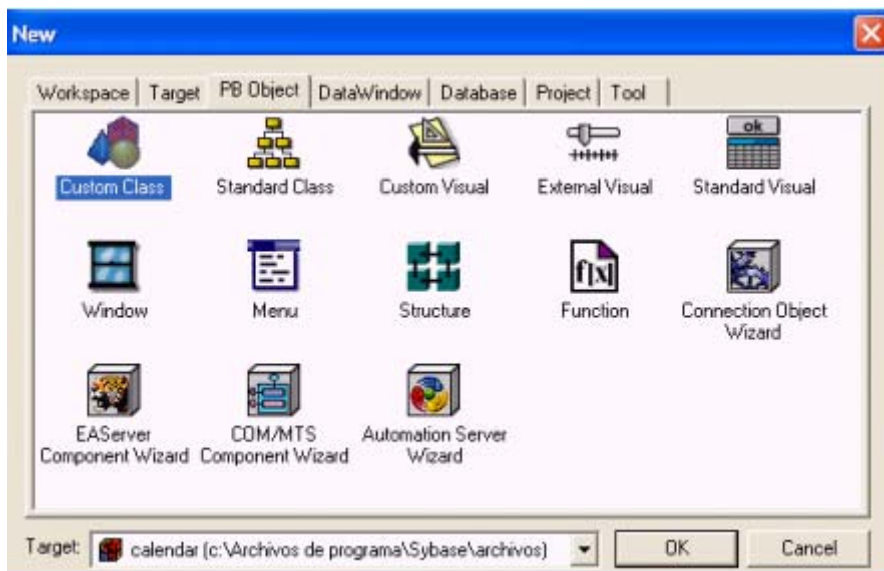
Dependiendo del tipo de aplicación que se desarrolla podemos definir los elementos que van a formar parte de la aplicación.

Enseguida se detalla algunas sugerencias para comenzar la programación con posibles objetos.


- Que tipo de arquitectura va ha tener la aplicación?  
Monopuesto, Cliente/Servidor o Distribuida.

- Realizar una creación de una nueva librería de clases o ya se la tiene desarrollada
- El interfaz de la aplicación está enfocado hacia una visualización en internet o hacia los elementos de PowerBuilder.

Al comenzar a crear cualquier proyecto debemos primero generar el objeto Workspace, según la jerarquía que anteriormente se mencionó. Esto se lo hace por medio del menú File → opción New → Workspace y digitando el nombre del mismo (extensión .wks). Luego se procede a crear la clase aplicación de la misma forma que el anterior objeto (extensión (.pbl), accediendo a la pestaña Application. Una vez creados estos dos objetos indispensables procedemos a escoger los distintos objetos de Powerbuilder (custom class, window, custom visual, menu, function, entre otros) que se integran en el interfaz del proyecto, accediendo a la pestaña PB Object.



**Figura 6.** : Ventana MDI con los diferentes painter.

Visualizamos las librerías creadas junto a sus componentes (application, window, etc) haciendo clic en el icono library  en la barra de herramientas Powerbar1.

Una particularidad de la clase Application es que podemos utilizarla para dos acciones:

1. Configuración de las propiedades del proyecto actual

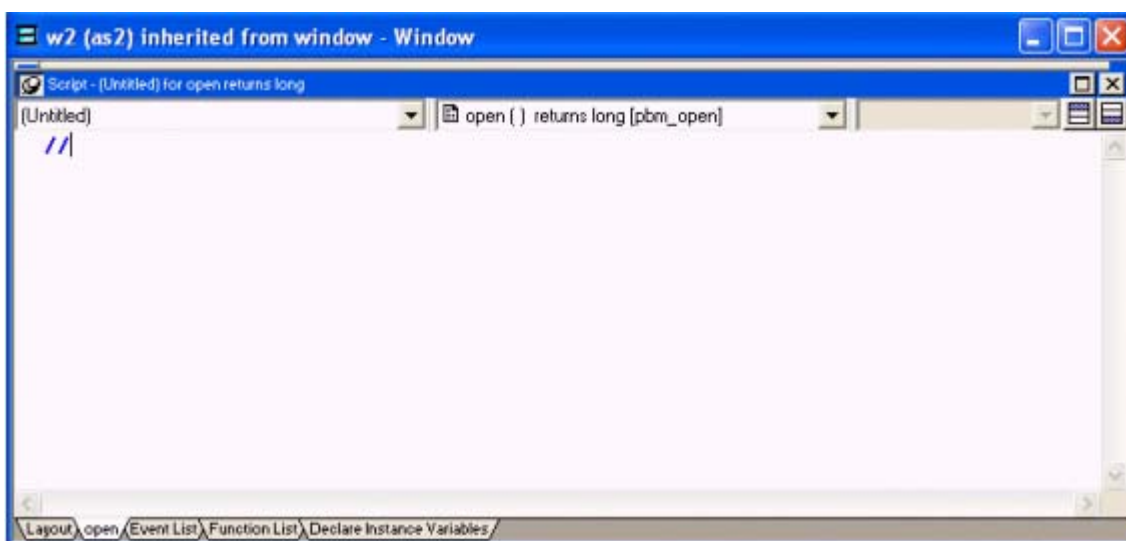
## 2. Generación de código

El punto 1 va a afectar a todo lo que contenga el proyecto, por tanto se trata de una acción global. Las propiedades más comunes son: tipos de letra, icono de la aplicación, variables globales, lista de todas las librerías.

### Herramienta de codificación

PowerBuilder posee una herramienta para realizar la codificación y se trata del lenguaje powerscript. Tiene dos entornos para introducir código, uno para funciones y el otro para eventos.

Powerscript no posee instrucciones para el acceso a datos externos. Para ello PowerBuilder permite el ingreso de sentencias de SQL embebido, aunque también se puede utilizar el objeto datawindow para el mismo propósito.



**Figura 6.3. Editor de código (PowerScript)**

## 6.4 Diseño previo del Aplicativo

La interfaz que se le presenta al usuario debe ser simplificada y no debe abarcar muchos detalles porque se puede convertir en una interfaz compleja, por tanto, se procedió a diseñar manualmente todas las

ventanas y objetos que se visualizaran en pantalla. El formato de todas las ventanas se basa en el siguiente diseño.

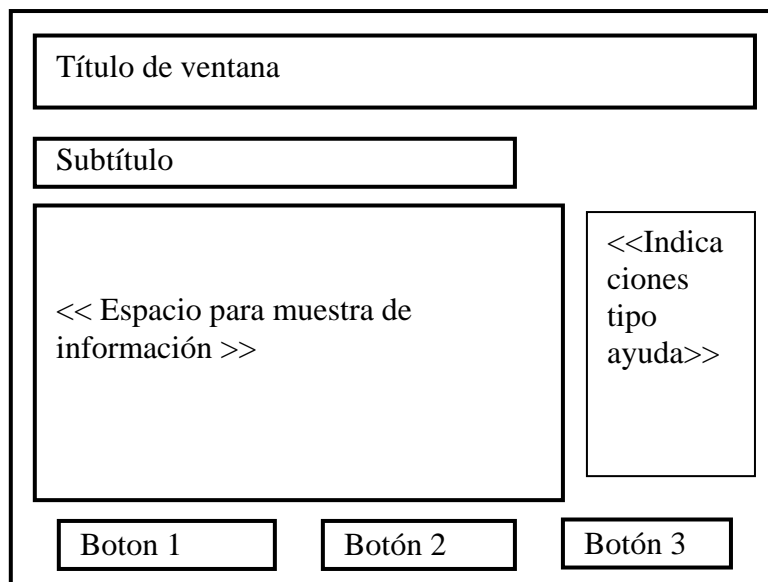


Figura 6.4. Diseño de las ventanas del sistema

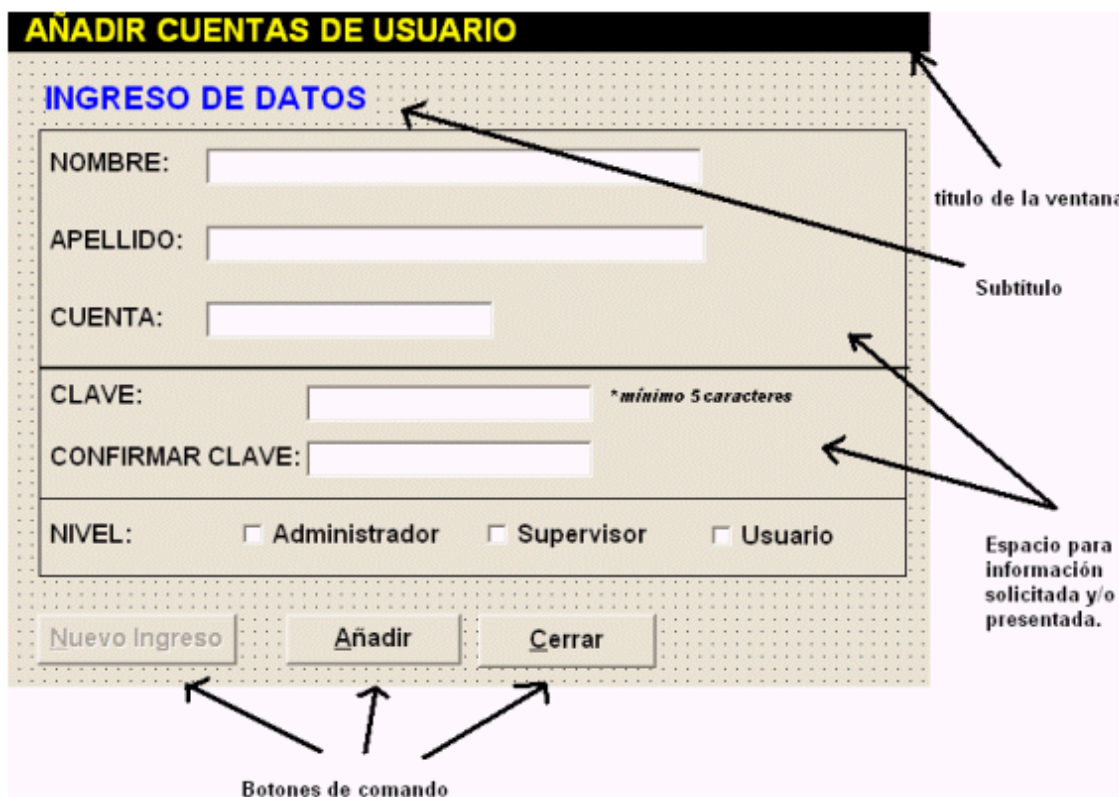
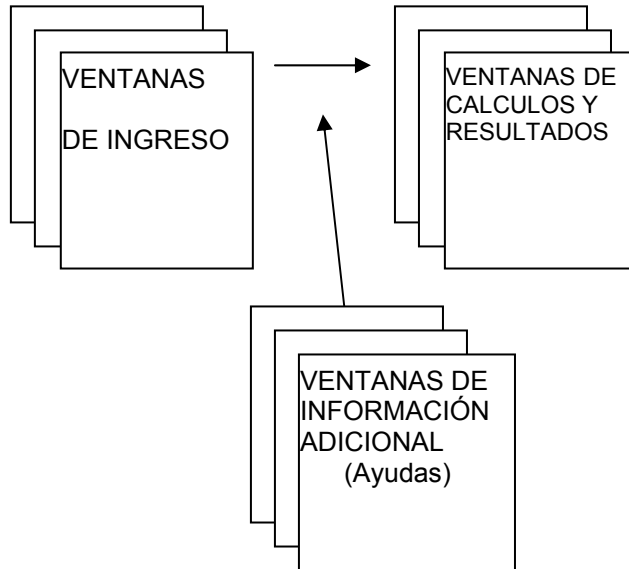


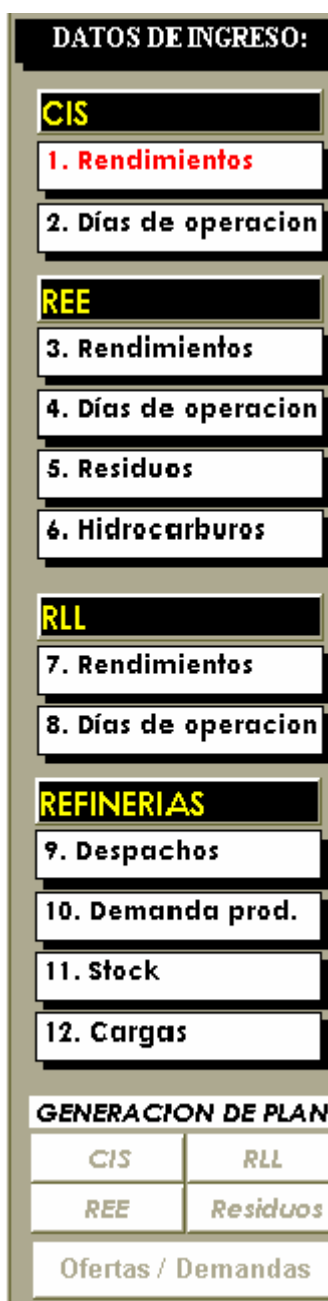
Figura 6.5. Ejemplo del tipo de ventanas a visualizarse

El orden en que el usuario va a visualizar las pantallas queda definido de la siguiente manera.



**Figura 6.6.** Orden de las ventanas del sistema

Para el caso del plan de producción las ventanas de ingreso son accesibles desde una lista de botones de comando.



**Figura 6.7.** Botones para el acceso a las ventanas de ingreso

### 6.5 Detalle de las fases de avance

Existen cuatro fases de avance o desarrollo de un sistema, las cuales se han aplicado a este aplicativo siguiendo el orden establecido. Estas fases son: Planificación, Diseño, Desarrollo y Mantenimiento.

Entendiendo por **planificación**, la fase donde se definen requerimientos funcionales y técnicos del sistema (hardware y software) junto con la rentabilidad del sistema y el plan de ejecución.

Se definen los elementos y sus funciones que intervienen en el sistema, entre ellos están:

- Estudio y análisis de los términos del área de petróleo.
- Revisión de la información de crudos que se encuentran almacenadas en hojas de cálculo (MS Excel).
- Estudio y comparación de las distintas versiones de las herramientas de programación.
- Definición preliminar de formularios presentes en la interfaz.

Por otra parte la fase de **diseño**, permite establecer la funcionalidad del sistema, por medio de la visualización de todos los procesos que intervienen. Además se determina el diseño técnico, donde se toma en cuenta la arquitectura final que va a tener el sistema.

Se realiza las siguientes tareas:

- Diseño de los diagramas de objetos
- Diseño y generación de la base de datos para los crudos
- Comprobación de la integridad y consistencia de la base de datos
- Diseño de la interfaz de usuario.
- Verificación de los vínculos de elementos de la interfaz.

La fase de **desarrollo** o también conocida como la de programación y pruebas, está enfocada a dar funcionalidad a la anterior fase, es decir codificar todos los objetos integrados al interfaz, además de incorporar aspectos de seguridad, normativas, documentación y otros requisitos. Luego se procede a la respectiva evaluación para comprobar la debida funcionalidad de los elementos del aplicativo en mención.

Por lo tanto, se revisa enseguida los procedimientos realizados para la obtención de los resultados esperados.

### **Formulario: Cargas a las refinerías**

#### **Procedimiento:**

- Los datos de ingreso de cargas son los mismos que van al reporte sin modificación alguna.
- Para el total de producción en una refinería, se realiza la operación de suma de todas las cargas de las unidades de esa refinería, así:

$$\text{Total\_carga (barriles/día)} = \sum \text{cargas de cada unidad}$$

- El total anual de carga de una refinería se define así:

$$\text{Total\_carga\_anual} = \text{Total\_carga} * \text{días\_operación}$$

### **Formulario: Plan de producción**

#### **Procedimiento:**

- Para cada mes, las filas que indiquen carga, se calcula, así:  
 $\text{carga\_refinería} = \text{carga\_unidad1} * \text{días\_operación} + \text{carga\_unidad2} * \dots$
- Para los derivados generados en un plan de producción

$$\text{Derivado} = \text{rendimiento} * \text{carga} * \text{días\_operación}$$

- Para la producción de los derivados, se utiliza la fórmula:

$$\text{Derivado1} = \text{porcentaje\_despacho} * \text{demanda\_derivado}$$

- Hay propiedades que dependen de otras, como la producción de diesel 2, que se define así:

$$\text{Producción\_diesel2} = \text{producción\_total} - \text{diesel1} - \text{diesel\_premium}$$

$$\text{Nafta} = \text{rendimiento\_nafta} * \text{carga\_unidad}$$

### **Formulario: Demandas**

#### **Procedimiento:**

- El total de demanda de cada derivado se establece así:  
 $\text{Demanda\_derivado} = \text{rendimiento\_derivado} * \text{producción\_derivado}$
- Todas las demandas de los derivados producidos por las tres refinerías son detallados en un solo formulario.



- Son valores que se registran mensualmente.

Además de estos formularios se hallan otros, en los que se aplica la misma lógica en cuánto al cálculo; esto es, por medio de fórmulas.

Para la generación de los resultados de algunos formularios se necesita de datos de uno o varios otros formularios, por lo que el aplicativo está debidamente organizado para que no haya falla en acceso a datos.

Como última y muy importante fase es la de **mantenimiento**, donde se revisa y se implementa mejoras funcionales al sistema, debido a que puede presentarse elementos externos que afecten al sistema y por tanto éste necesita de una actualización (en caso de cambios de parámetros en la red).

Existen tres tipos de mantenimiento determinados para un sistema informático, y son: mantenimiento correctivo o preventivo, adaptativo y perfectivo. En el correctivo se da prioridad a problemas de mal funcionamiento del sistema; en el adaptativo se soluciona problemas por modificación de elementos externos pero integrados al sistema; y, en el perfectivo se toma en cuenta aspectos de actualización y/o mejoras que se puedan adicionar a la fase de diseño.

Se detalla a continuación todas las etapas ejecutadas para la realización del aplicativo, las mismos que están enfocados a ejecutar operaciones específicas. Estas etapas están incluidas en las respectivas fases que antes se revisó. Cabe indicar que se enumeran las etapas de avance sólo del frontal de la aplicación, ya que la parte del desarrollo de la base de datos ya se mencionó en el capítulo anterior.

- 1. Determinación de herramientas de programación.-** El software utilizado para el desarrollo del todo el sistema es: PowerBuilder y PowerDesigner de Sybase, SQL Server de Microsoft y otras herramientas adicionales.

Además se toma en cuenta los requerimientos de hardware que están detallados en el Capítulo 5 – Sección: Requisitos de Hardware.

2. **Creación de los objetos o elementos iniciales.-** La integración de elementos que conformaran el interfaz, es el primer paso para el desarrollo del sistema. Se realiza la creación de Workspace, Application, Window, Menu, etc.
3. **Diseño de la Interfaz.-** Presentación preliminar del diseño de las ventanas, aplicando así las diferentes propiedades y atributos: tamaño, color, ubicación, tipo de letra, estilos, efectos, etc.
4. **Funcionalidad de los elementos.-** Todos los controles y objetos implementados en la interfaz del sistema han sido codificados para que ejecuten sus apropiadas funciones. Así, se comenzó a dar funcionalidad a las opciones del menú, botones, cuadros de diálogo, etc.
5. **Evaluación del sistema.-** Se realiza varias pruebas del sistema, mediante el ingreso de datos, verificando la funcionalidad de los elementos o controles del interfaz, modificando parámetros de entrada y salida de información, comprobando aspectos de seguridad de ingreso, etc.
6. **Documentación.-** En esta sección se detalla todo lo referente al diseño, desarrollo e implementación del sistema, comenzando por los requerimientos de hardware y software, elementos del diseño de la base de datos, elementos del diseño del interfaz, detalle de las funciones de cada objeto del interfaz, codificación de los elementos del interfaz (funciones, procedimientos, eventos), etc.
7. **Mantenimiento.-** Con el fin de actualizar el sistema a nuevas versiones de la plataforma Windows se esta tomando en cuenta los parámetros que deberán cambiar para el efecto. Además se revisa

las configuraciones de la red en caso de que se necesite registrar a otros servidores y/o clientes.

## 6.6 Pruebas y evaluaciones

Las pruebas constan de varias ejecuciones del sistema, ingresando y procesando operaciones para ver y analizar la confiabilidad de los resultados finales. El modelo de las ventanas de ingreso de datos se muestra en la siguiente figura, donde están los controles y objetos apropiados para la entrada de información.

The image shows a software interface with a yellow header bar containing the text "RENDIMIENTO (% vol)" and "COMPLEJO INDUSTRIAL SHUSHUFINDI". Below the header, there are two main panels. The left panel is titled "COMPLEJO INDUSTRIAL SHUSHUFINDI" and contains a table with the header "PRODUCTOS \ RENDIM." and several rows. The right panel is titled "PLANTA DE GAS DE SHUSHUFINDI" and contains a table with the header "PRODUCTO \ RENDIM." and several rows. Between the two panels, there are labels: "PROMEDIO DE CARGA", "RENDIMIENTOS P.GAS", "G. NATURAL A AMZ (BL)", and "FACTOR DE CALCULO:". The interface is designed for data entry into a simulation model.

Figura 6.8. Modelo de las ventanas de ingreso de datos

## 6.7 Comprobación de resultados de la simulación con mezclas de crudos.

Para ver los resultados previo el ingreso de los respectivos datos, se ha incorporado un conjunto de botones que generan los respectivos planes de producción de las refinerías, los mismos que al pulsarlos presentan la ventana de resultados, que tiene el siguiente formato:

PLAN DE PRODUCCION		COMPLEJO INDUSTRIAL SHUSHUFINDI										
CARGA DIARIA: <input type="text"/>												
CONCEPTO	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
PLANTA DE GAS SHUSHUFINDI												
CARGAS												
	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
GAS NATURL(BLS)												
LICUABLES(BLS)												
TOTAL												
GAS NATUR(MMPC)												

Figura 6.9. Modelo de la ventana de resultados.

## 6.8 Correcciones

Las correcciones que se han ejecutado están relacionadas a la codificación de los elementos de la interfaz, ya que existieron errores de diferente tipo, tales como de operaciones matemáticas (divisiones para cero), de dimensión de las matrices (sobredimensionamiento o desbordamiento), de transcripción de fórmulas, de tipos de datos (asignación de enteros a cadenas y viceversa), accesos a datos erróneos, cálculos con datos nulos, procedimientos con infinitas iteraciones, de conexión al servidor de base de datos, entre otros.

Aunque es importante mencionar que también se efectúan correcciones a nivel de interfaz, iniciando con la reubicación de los elementos de la interfaz, la aplicación de las propiedades y atributos de cada elemento, y así añadiendo características importantes para dar una mejor comprensión de lo que se visualiza en pantalla.

## 6.9 Implementación del sistema

Una vez finalizado el sistema, se procede a la instalación en los equipos pertinentes, para esto se toma en cuenta las consideraciones de hardware mencionadas en el capítulo anterior, tanto para servidor como para cliente.

1. Se inicia instalando el administrador de base de datos SQL Server con el fin de generar la base de datos para crudos en el equipo Servidor. Para el caso de la empresa Petroindustrial el servidor se encuentra en un laboratorio distante del equipo Cliente.
2. Una vez instalada la base de datos se procede a ejecutar el script de la base de datos para crudos que ha sido generado con la herramienta case PowerDesigner; este script se lo puede ejecutar desde la utilidad Analizador de Consultas de SQL Server.
3. Se procede a la instalación del software de base de datos en el equipo cliente con las opciones respectivas. Aquí se instalan las herramientas necesarias para la conexión entre el equipo servidor y cliente.
4. Creación de una fuente de datos de sistema (DSN de sistema) para la conexión a la base de datos vía ODBC.
5. Instalación del sistema de crudos, que se lo realiza mediante el fichero ejecutable que ha sido creado para el efecto.

## 6.10. Documentación

La documentación de un sistema implica la ejecución de diferentes etapas como son: recopilación de información, análisis de la información, elaboración de manual o guía tanto técnica como de usuario, entre otras etapas.

La parte de documentación técnica sirve para posibles actualizaciones o modificaciones que se deban realizar por parte del o los programadores. Por otro lado la documentación como guía de usuario está enfocada a los usuario que interactúan directamente con el producto terminado.

En la documentación de este sistema de crudos consta los siguientes puntos: los tres diagramas UML básicos (casos de uso, clases, secuencia), diseño de los módulos y su codificación, diseño del interfaz, ejemplos y recomendaciones.

### **6.11 Consideraciones en la implementación del aplicativo**

Una vez concluido el aplicativo, quedará para uso de los respectivos departamentos de operación de Petroindustrial, con el fin de que lo evalúen y saquen reportes actualizados de las operaciones de refinería.

Este aplicativo realizará los siguientes tipos de reporte:

- Cargas de todas las unidades de proceso de las tres refinerías (Ver Capítulo 5, Figura 5.9), junto con su porcentaje de funcionamiento.
- Días de operación en cada mes de las unidades de proceso.
- Producción de los derivados de cada refinería
- Stock de los derivados de petróleo
- Oferta y Demanda de los derivados de petróleo
- Rendimientos de cada derivado
- Cálculo de Residuos en la Refinería Estatal Esmeraldas
- Cálculo de la producción de gasolinas

Estos resultados finales son un resumen mensual o anual de las operaciones que se realizan en refinería con los productos obtenidos a partir de los crudos.

A futuro, se pretende guardar este tipo de información en archivos físicos u otro tipo de soporte en el que se pueda leer con facilidad en cualquier computador sin la necesidad de instalar algún paquete adicional. Esta aclaración se debe a que, actualmente los datos se encuentran almacenados en una base de datos, que cuenta con seguridades de acceso a dicha información.

Además de lo anteriormente expuesto; es conveniente indicar que este aplicativo permitirá reducir el tiempo que se invertía en calcular las operaciones de refinería mencionadas, ya que, en realizar un plan de producción tomaba cerca de dos semanas.