

CAPITULO V

CONSTRUCCION DEL SOFTWARE

Utilizando la modelación orientada a objetos, estructuramos el capítulo de *Construcción del Software* en las fases siguientes:

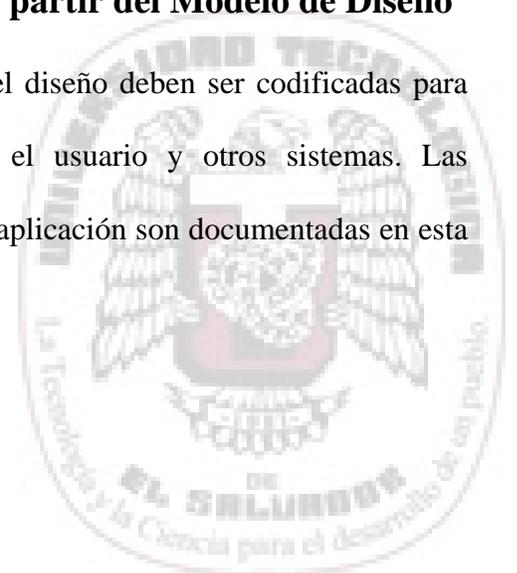
- Implementación de Código a Cada Clase a Partir del Modelo de Diseño.
- Integración de Clases Extraídas de Biblioteca de Reutilización.
- Construcción del Prototipo o Incremento.
- Identificación de Responsabilidades y Colaboraciones.

El Análisis Orientado a Objetos (AOO) y Diseño Orientado a Objetos (DOA) nos proporcionan una base bien definida para construir el software en Programación Orientada a Objetos (POO) con sus diversos requerimientos y optimizaciones.

A continuación presentamos las fases de la construcción de software.

5.1 Implementación de código a cada clase a partir del Modelo de Diseño

Cada una de las clases y subclases identificadas en el diseño deben ser codificadas para proporcionar una interfaz que tenga relación con el usuario y otros sistemas. Las estructuras de los programas mas sobresalientes en la aplicación son documentadas en esta sección.



Estructura de programa para el acceso del sistema

Declaracion de variables generales y conexión con la base de datos tipo ODBC

Option Explicit

Dim (Declaración de variables generales)

Ejemplo:

Dim cnBASERRHH As ADODB.Connection

Dim RsBancoCurriculos As ADODB.Recordset

Private Sub cmdCancel_Click()

Procedimiento para suspender entrada a la aplicación (Botón Cancelar)

Instrucción: End

End Sub

Private Sub cmdOK_Click()

Operación para ejecución del formulario que presenta el menú principal.

Llama al procedimiento de validación del pasword.

Ej. MDIPrincipal.Show

Solo se ejecuta al validad al usuario en la caja de texto "Password"

End Sub

Private Sub Form_Load()

Procedimiento para establecer conexión con la base de datos ODBC

Ejemplo:

Set cnBASERRHH = New ADODB.Connection

Set RsBancoCurriculos = New ADODB.Recordset

cnBASERRHH.CursorLocation = adUseServer

cnBASERRHH.Open "dsn=BASERRHH"

RsBancoCurriculos.CursorType = adOpenDynamic

RsBancoCurriculos.LockType = adLockBatchOptimistic

RsBancoCurriculos.Source = "Select *From BancoCurriculos order by
NombreAspirante"

Set RsBancoCurriculos.ActiveConnection = cnBASERRHH

RsBancoCurriculos.Open

End Sub

Private Sub txtUserName_Change()

'Procedimiento para habilitar botón de acceso

If txtUserName.Text <> " " And txtPassword.Text <> " " Then

cmdOK.Enabled = True

Else

cmdOK.Enabled = False

End If

End Sub



```
Private Sub Buscar()  
Operación de búsqueda del registro de claves de acceso  
Toma el valor de la clave de acceso y busca en la tabla de datos.  
End Sub
```

Estructura de programa para menú de la aplicación

```
Private Sub Banco_Click()  
Operación para ejecutar primera opción del menú  
Ejemplo:  
FrmBancoDatos.Show  
End Sub
```

```
Private Sub Empleados_Click()  
Operación para ejecutar segunda opción del menú  
Ejemplo:  
FrmEmpleados Show  
End Sub
```

```
.  
. .  
Operaciones de ejecución por cada opción del menú  
. .
```

```
Private Sub MDIForm_Load()  
Proceso general del objeto menú.  
End Sub
```

```
Private Sub Salir_Click()  
Proceso de salida del objeto menú  
End (del menú)  
End Sub
```

Estructura de programa para mantenimiento de una clase tipo entidad

```
Declaraciones generales y de conexión  
Option Explicit  
Ejemplo:  
Dim cnBASERRHH As ADODB.Connection  
Dim RsBancoCurriculos As ADODB.Recordset  
Dim BlnEditar As Boolean
```



Private sub navegar
Procedimiento de llamada a clase reutilizable de navegación de registros
 Ir a registro inicial
 Ir al registro final
 Ir al registro siguiente
 Ir al registro anterior
EndSub

Private Sub CmdBuscar_Click()
Operación para buscar un registro.
End Sub

'Procedimiento para llamada de clase reutilizable
 Cancelar operación
 Guardar registro

Private Sub CmdEliminar_Click(Index As Integer)
'Proceso de eliminar un registro

Private Sub CmdGuardar_Click()
Proceso para guardar el registro editado o nuevo
Endsub

Private Sub CmdImprimir_Click()
Operación que permite Imprimir lista de registros
End Sub

Private Sub CmdModificar_Click(Index As Integer)
Operación que permite modificar un registro
EndSub

Private Sub CmdNuevo_Click(Index As Integer)
Operación que permite crear un nuevo registro
End Sub

Private Sub cmdSalir_Click(Index As Integer)
Procedimiento en botón de salir
 Unload Me
End Sub

Llamada a clase reutilizable de desactivación del menú de mantto.

Llamada a clase reutilizable de activación del menú de mantto.



Private Sub Form_Load()

Procedimiento de carga del formulario con las especificaciones de conexión con la base de datos

Private Sub cmdDeshacer_Click()

Procedimiento para anular efecto de modificado

End Sub

Private Sub Form_Unload(Cancel As Integer)

Procedimiento para cerrar la base de datos

Ejemplo:

cnBASERRHH.Close

Set cnBASERRHH = Nothing

End Sub

Private Sub limpiar()

Procedimiento para limpiar las cajas de entrada de datos

End Sub

Private Sub MostrarDatos()

Operación de despliegue de datos del registro

End Sub

Private Sub Bloquear()

Procedimiento para bloquear acceso a los datos del registro

End Sub

Private Sub DesBloquear()

Procedimiento para permitir acceso a los datos del registro

End Sub

5.2 Integración de Clases Extraídas de Biblioteca de Reutilización

La biblioteca de clases es formada por todas aquellas clases que pueden ser reutilizadas.

Entre ellas podemos mencionar:

- Navegación en la tabla de datos.
- Guardar registro o cancelar operación.
- Consultar un registro de empleado.
- Consultar un expediente de aspirante a plaza.



- Búsquedas entre un rango de fechas.
- Mensaje de advertencia de eliminación.
- Mensaje de pausa en despliegue de datos.
- Mensaje de respuestas S/N.

5.3 Construcción del Prototipo o Incremento

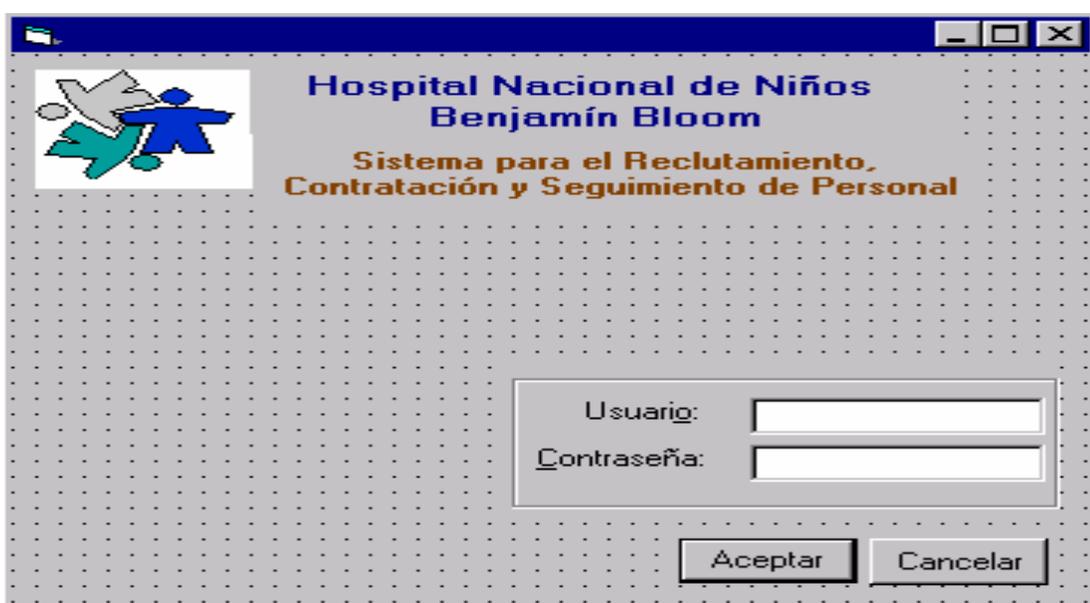


Figura No.1

El acceso al sistema esta definido por un objeto que permite ingresar solo si existe la clave de acceso autorizada. Cada usuario tiene una clave que tienen una asignación de derechos según sus funciones. Las funciones están definidas por el subsistema al que tiene acceso un usuario.



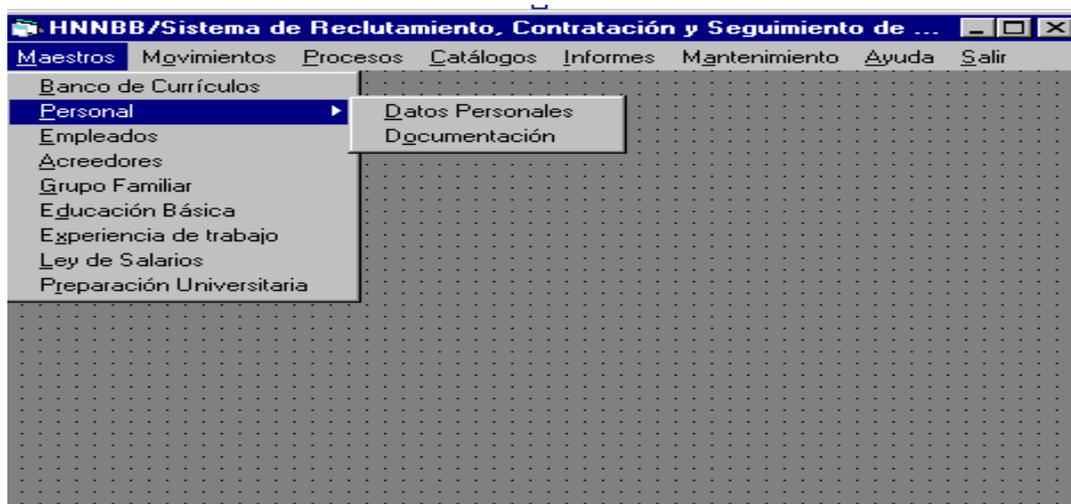


Figura No.2

Las opciones bajo Maestros permiten el acceso a tablas que contienen datos relacionados con las entidades principales de la aplicación. Estas tablas generalmente son actualizadas por registros de movimientos y participan en todas las cargas de datos de la aplicación.

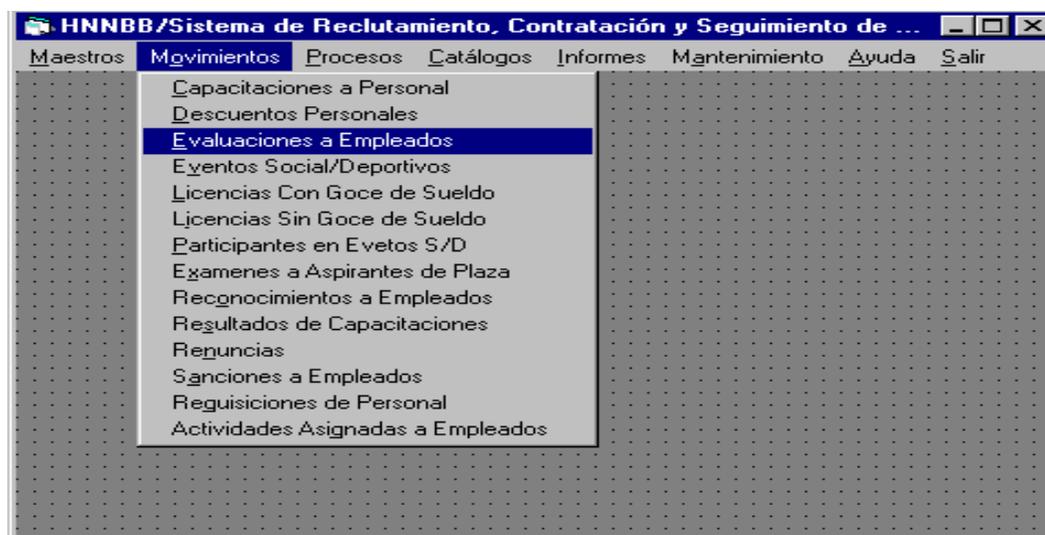


Figura No.3

Los registros de movimientos son todos aquellos que crecen constantemente por cada gestión administrativa del empleado.

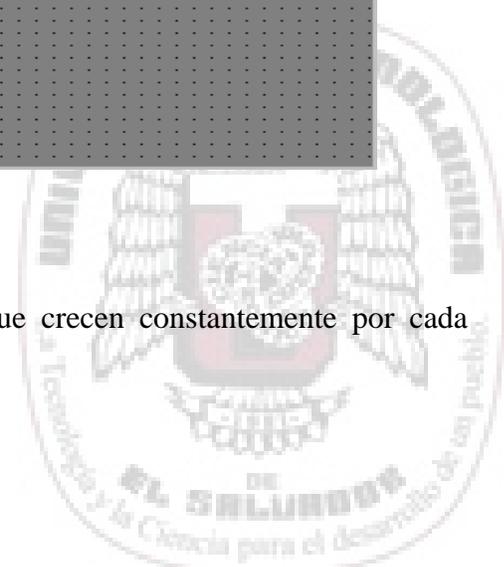




Figura No. 4

Los procesos de actualización o generación de registros consolidados son ejecutados desde la opción de procesos. Esto incluye la creación de apertura anual donde se crean nuevas tablas maestras.

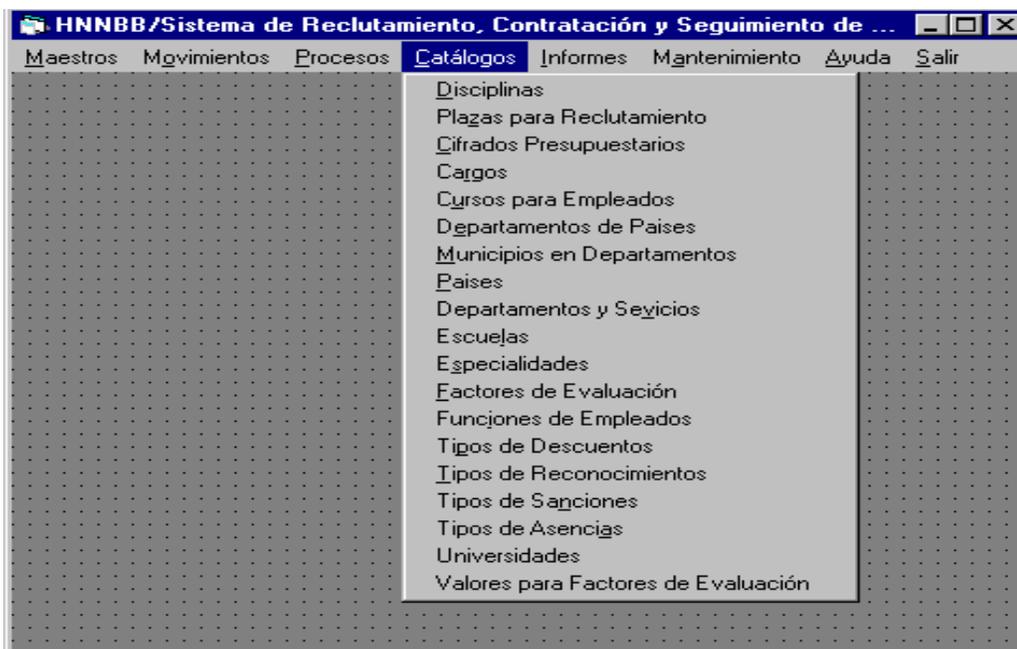
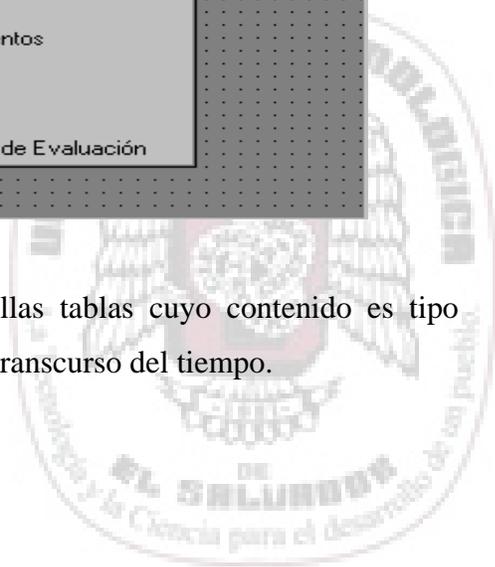


Figura No. 5

La opción catálogo presenta el acceso a todas aquellas tablas cuyo contenido es tipo catálogo o que no reciben cambios significativos en el transcurso del tiempo.



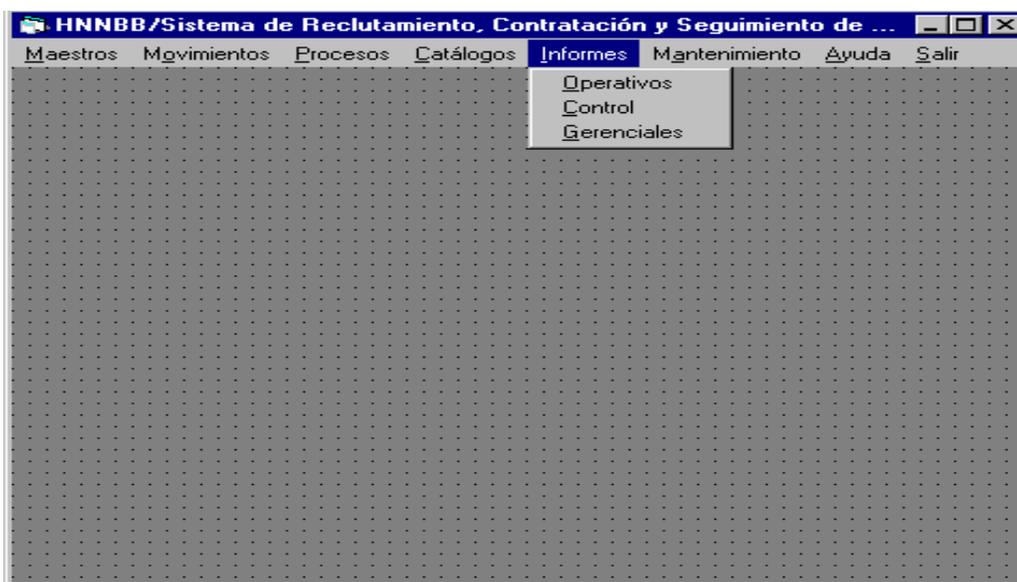


Figura No 6

Los Informes/Consultas se dividen en tres tipos: Operativos (detallados), de Control (semi-detallados) y Gerenciales (resúmenes y estadísticos).

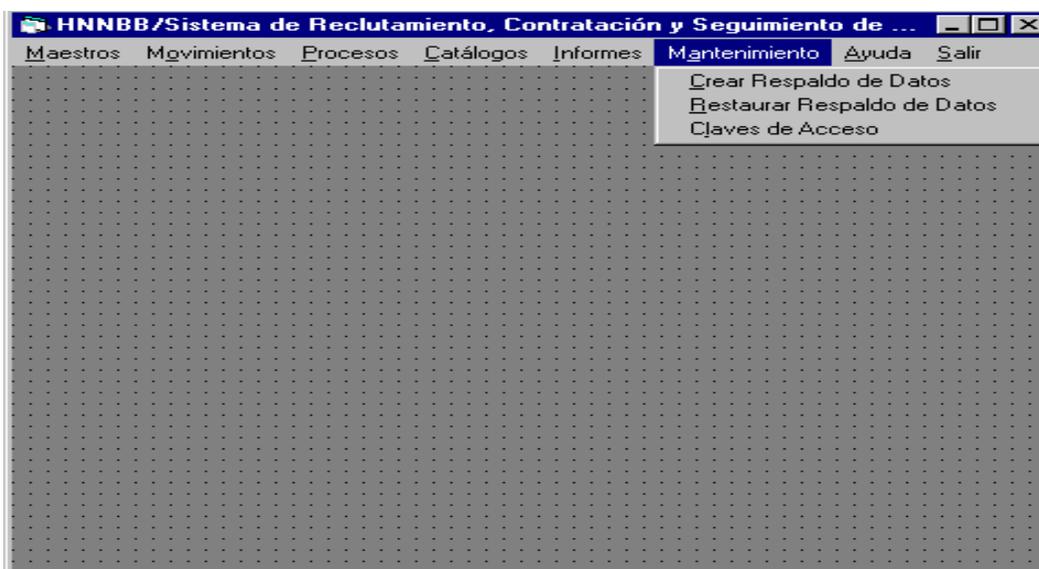
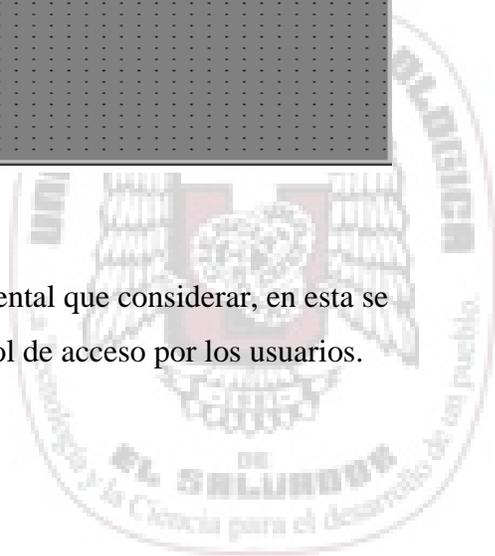


Figura No. 7

El mantenimiento de la aplicación es una parte fundamental que considerar, en esta se incluyen la creación de respaldo de los datos y el control de acceso por los usuarios.



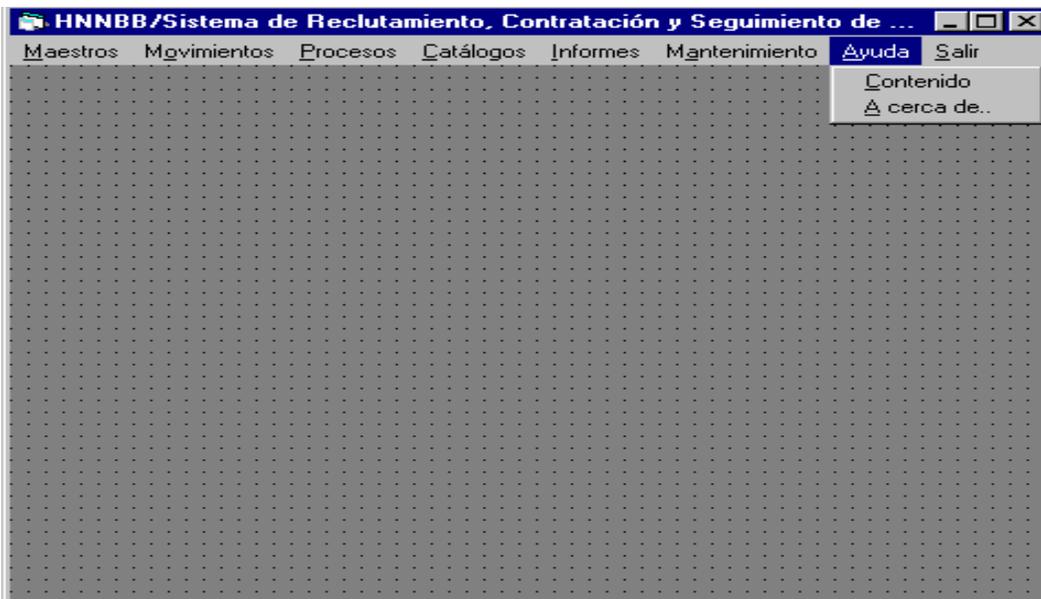


Figura No. 8

Por ultimo tenemos la opción de ayuda del sistema. En esta podemos consultar al manual del usuario y otra información relacionada con el uso del software.

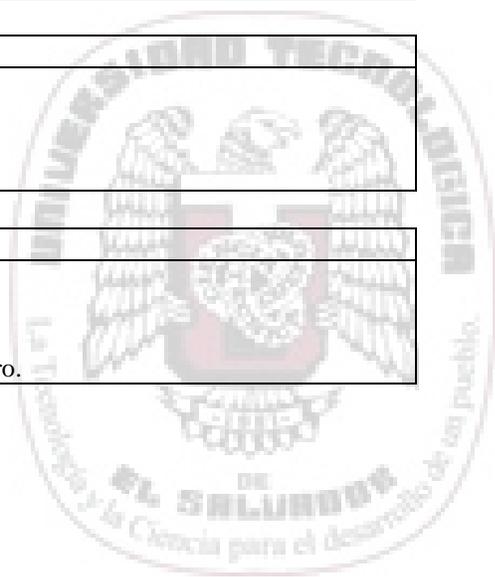
5.4 Identificación de Responsabilidades y Colaboraciones

Nombre de la clase: NavegaciónReg
Responsabilidades Operaciones: Mover puntero dentro de la tabla de datos.
Colaboraciones o relaciones entre clases -Se interrelaciona con todas las clases y objetos que permiten el mantenimiento de una tabla de datos.

Nombre de la clase: AdicionarReg
Responsabilidades Operaciones: Captura los datos en los objetos de datos.
Colaboraciones o relaciones entre clases -Permite que se active la clase GuardarReg para completar la acción de agregar un registro.

Nombre de la clase: GuardarReg
Responsabilidades Operaciones: Almacenar información en la tabla de datos.
Colaboraciones o relaciones entre clases -Complementa la acción de la clase AdicionarReg

Nombre de la clase: EliminarReg
Responsabilidades Operaciones: Eliminar un registro.
Colaboraciones o relaciones entre clases -Permite activar la clase AdverElimina antes de eliminar un registro.



Nombre de la clase: AdverElimina
Responsabilidades Operaciones: Advertir al usuario sobre la eliminación del registro.
Colaboraciones o relaciones entre clases -Complementa a la clase EliminarReg.

Nombre de la clase: ModificarReg
Responsabilidades Operaciones: Modificar la información existente en un objeto dato.
Colaboraciones o relaciones entre clases -Se complementa con la clase GuardarReg.

Nombre de la clase: BuscarReg
Responsabilidades Operaciones: Realizar búsqueda en una tabla en base a parámetros .
Colaboraciones o relaciones entre clases -Se interrelaciona con la clase PedirDato.

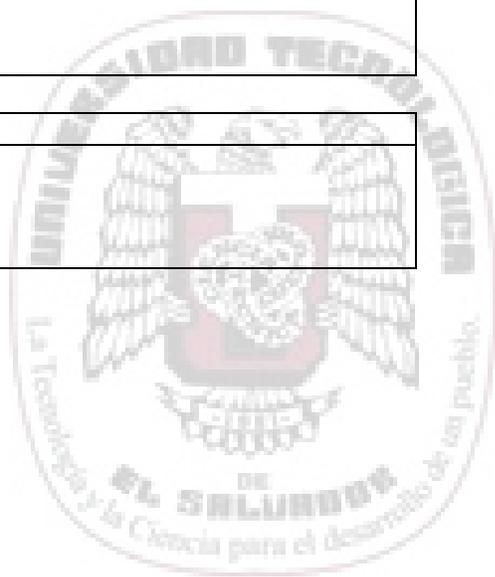
Nombre de la clase: PedirDato
Responsabilidades Operaciones: Permite pedir un dato tipo llave de registro para localizar un registro.
Colaboraciones o relaciones entre clases -Complementa a la clase BuscaReg

Nombre de la clase: ImprimirReg
Responsabilidades Operaciones: Despliega en pantalla e impresor la lista los registros de una tabla.
Colaboraciones o relaciones entre clases -Se complementa con la clase ConfigImp.

Nombre de la clase: ConfigImp
Responsabilidades Operaciones: Configura la impresión del informe.
Colaboraciones o relaciones entre clases -Colabora con las clases relacionadas a la impresión de informes.

Nombre de la clase: ConsulEmpleado
Responsabilidades Operaciones: Pedir el nombre del empleado.
Colaboraciones o relaciones entre clases -Complementa a la clase BuscarReg.

Nombre de la clase: ConsulExpe
Responsabilidades Operaciones: Pedir el número de un expediente de aspirante.
Colaboraciones o relaciones entre clases - Complementa a la clase BuscarReg.



Nombre de la clase: ConsulMov
Responsabilidades Operaciones: Consulta en base a un rango de fechas.
Colaboraciones o relaciones entre clases -Se complementa con la clase configImp .

Nombre de la clase: Mensaje
Responsabilidades Operaciones: Escribe todos los mensajes en la aplicación.
Colaboraciones o relaciones entre clases -Se relaciona con todas las clases de la aplicación.

Este capítulo permitió documentar la construcción del software con sus diversas partes elementales. En la técnica de desarrollo orientado a objetos la fase siguiente debe ser “las pruebas del software”. El próximo capítulo muestra en detalle todos los componentes que se deben considerar.

