CAPÍTULO VI

REFERENCIA TÉCNICA Y OPERATIVA DEL PROTOTIPO DE ADMINISTRACIÓN DE NÓMINA

Este capítulo presenta la referencia técnica y operativa del prototipo de Administración de Nómina, esta dividido en dos secciones: la Referencia Técnica y la Referencia Operativa, básicamente esta documentado con figuras y las referencias al código de programación se puede revisar en los anexos del cd-rom adjunto.

6.1. Referencia Técnica

En este apartado se presentará la referencia técnica del proyecto, pasando desde un vistazo a las herramientas utilizadas hasta la codificación del proyecto, tanto en lo que se refiere al diseño de base de datos como al diseño del prototipo del aplicativo.

6.1.1. Modelo Conceptual de Base de Datos

El modelo conceptual de datos corresponde al diseño de base de datos básico a partir del cual se puede obtener una representación física, esta base de datos inicial no representa las características de un DBMS en particular, al contrario, permite luego de su concepción escoger el mas adecuado de acuerdo a un estudio y análisis de requerimientos. Este modelo se encuentra en términos de diagramas Entidad-Relación (E-R)

El diagrama del modelo correspondiente al prototipo funcional se encuentra en el archivo nomina.cdm (conceptual data model) que esta grabado en el cd-rom adjunto.

6.1.2. Herramientas de diseño

La diagramación, control de errores, chequeo y administración general del proyecto en esta etapa se ha realizado utilizando ***Sybase Power Designer 6.1.3 DataArchitect Suite*** , trabajando específicamente con el ***DataArchitect*** y el ***AppModeler***, además se ha instalado un juego de drivers ODBC de la casa fabricante ***Intersolv***.

6.1.3. Nomenclatura

El proyecto de diseño se denomina ***proy_nomina*** , la base de datos lleva el nombre de ***nomina***, todos los nombres de las tablas que componen la base de datos tienen la estructura ***tNombreDeTabla*** donde ***NombreDeTabla*** corresponde al nombre de la tabla en la base de datos y la letra ***t*** corresponde a un prefijo de ***tabla***. La nominación de los atributos de las tablas tiene la estructura ***tab_nombredecampo*** donde ***tab*** corresponde a un prefijo formado por al menos las 3 primeras letras del nombre de la tabla o en su defecto al menos 3 letras descriptivas de la tabla cuando por alguna razón exista ambigüedad.

En la implementación del código fuente se utiliza la estructura ***prenombrecontrol*** donde ***pre*** corresponde a la notación para los controles visibles en Visual Basic que se detalla mas adelante y ***nombrecontrol*** corresponde al nombre del control como tal, algunos de los prefijos para los controles mas utilizados son:

Tabla 6.1

etq	etiqueta
cmd	comando
txt	cuadro de texto
rd	remote data
ado	ADO Data control
f	formulario
de	data environment
dr	data report

y en general, para otros controles se utiliza entre 1 y 4 letras descriptivas del mismo. La declaración de variables globales del proyecto junto con su respectivo funcionamiento se encuentra en el modulo de código ***mVarGlob***, indicando además que los archivos correspondientes al proyecto guardan así mismo una nomenclatura, tal como sigue:

Tabla 6.2

f	Archivo de formulario
dr	Archivo para un Data Report
de	Archivo para un Data Environment

6.1.4. Normalización

El diseño en su totalidad se encuentra bajo la tercera Forma normal 3NF, tal como se lo puede ver y revisar el archivo con los modelos tanto conceptual como físico (***nomina.cdm*** y ***nomina.pdm***) con la herramienta ***DataArchitect*** de ***Sybase Power Designer.***

6.1.5. Integridad Referencial

La base de datos en su totalidad responde a un diseño de actualización y eliminación bajo :

Tabla 6.3

Restricción	*Declarativa*
Cascada	*Trigger*
Set Null	*Trigger*
Set Default	*Trigger*

6.1.6. Independencia de datos

El modelo conceptual de la administración de nomina planteado fue creado con ***Sybase Power Designer***, esto sugiere que en esta instancia no se ha tomado en cuenta la implementación física de un administrador de base de datos. Posteriormente, el modelo conceptual ha sido generado para una implementación física, la misma que se muestra en el modelo físico del archivo nomina.pdm (***physical data model***, que se encuentra en el CD-ROM adjunto), es decir el presente diseño bien podría implementarse en cualquier DBMS y solamente se debería volver a crear un DSN hacia la nueva base de datos y la aplicación funcionaria correctamente demostrando que la implementación de la base de datos puede ser independiente de la programación del frontal, tomando en cuenta que en su totalidad los procesos del aplicativo se realizan en el lado del cliente bajo sentencias SQL básicas.

6.1.7. Modelo Físico de Base de Datos

El modelo físico de datos es el siguiente paso del diseño de Base de Datos, este consiste en representar el modelo conceptual con las características de un DBMS escogido previamente, en el caso del modelo del prototipo se ha escogido el DBMS ***Sybase Adaptive Server Enterprise 11.5*** con sus correspondientes características. El archivo que contiene el modelo físico se llama nomina.pdm y se encuentra en el CD adjunto, para visualizar el modelo se deberá utilizar la herramienta ***Power Designer AppModeler for Power Builder*** de ***Sybase***, a partir de este modelo se ha generado la base de datos física en el servidor ***Development*** y al mismo tiempo el script de generación de la misma para futuros requerimientos.

6.1.8. Instalación del Servidor y las herramientas de cliente

Al momento de insertar el CD de instalación de ASE Sybase System 11 debe ejecutarse un autorun que le solicita escoja una alternativa de instalación (ver figura 6.1).

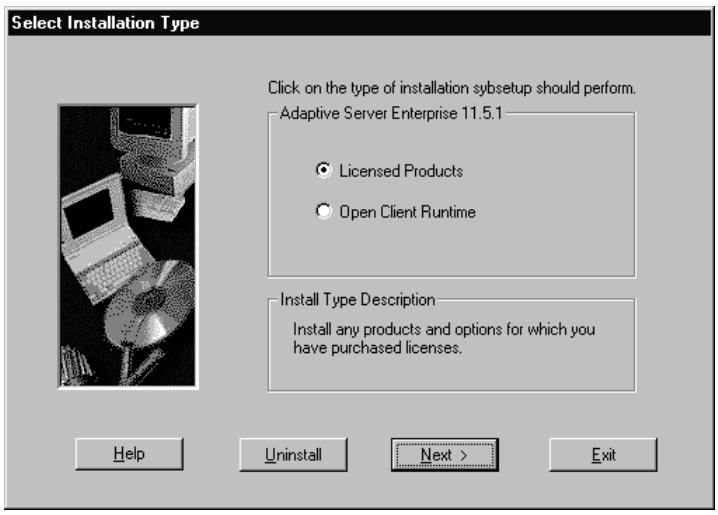


Figura 6.1

La opción ***Licensed Products*** instala todas las herramientas del servidor y el ***Sybase Central*** para administración de los servidores de bases de datos. Esta opción le permite crear el servidor de datos propiamente dicho en el Sybase Central (en este caso el servidor DEVELOPMENT), y al mismo tiempo crea los servidores de Backup, de

Monitoreo y el de Históricos. Junto con esto (a excepción de ***Power Designer 6.1.3 32 Bit***) se instalaran las opciones que se presentan en la figura 6.2 y que son parte del menú Sybase en el Menú Programas.

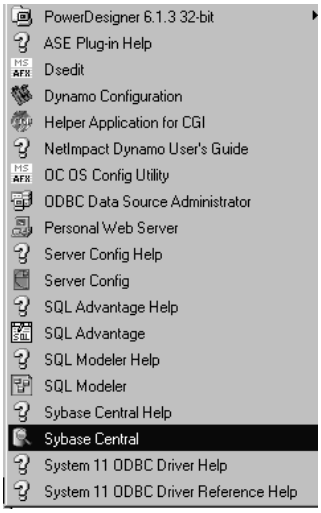


Figura 6.2

La opción ***Open Client Runtime*** instalara las herramientas y librerías de cliente en cada uno de los clientes (con sistemas operativos descritos en el apartado ***1.11.3 Plataforma de Servidor, Cliente y Red***). Básicamente instala la opción ***OC OS Config Utility*** (que se muestra en la figura 6.3) y algunos archivos de configuración de inicio (p.e. sql.ini) importantes para realizar la conexión del cliente. Además se instalaran los drivers ODBC del motor del DBMS ***Sybase System 11***.

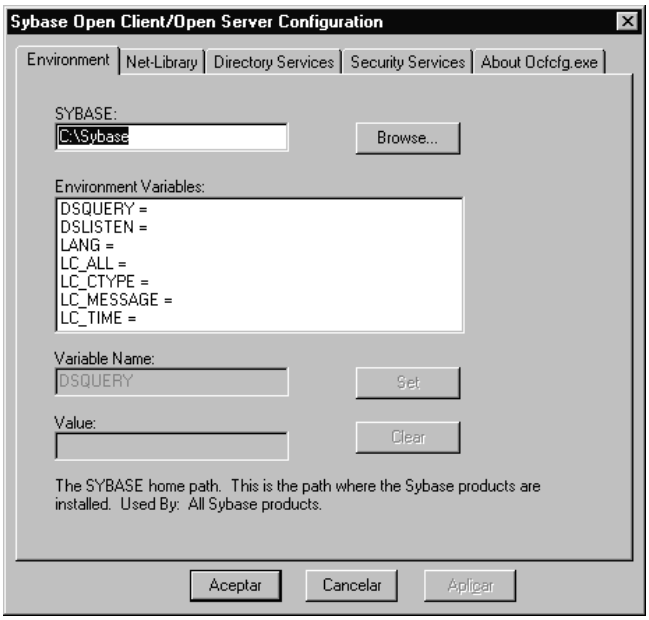


Figura 6.3

Cabe recalcar que para acceder al servidor SQL (DEVELOPMENT) con una herramienta de consulta SQL como ***Isql*** se debe realizar los siguientes pasos :

- Copiar el archivo sql.ini del directorio C:\Sybase\ini o su equivalente.
- Pegar sql.ini copiado en el directorio $:\Sybase\ini del cliente.
- Adicionar la siguiente línea marcada en el sector [SQLSERVER] del archivo win.ini :

  [SQLSERVER]

  AutoAnsiToOem=off

  DEVELOPMENT=dbmssoc3,192.168.0.22,5000

  SUPPORT=dbmssoc3,192.168.0.117,5000

  Nótese que la sintaxis es como sigue; DEVELOPMENT es el nombre del servidor de datos en el Sybase Central , seguido del servicio de conexión de base de datos **dbmssoc3**, luego la **dirección IP** del servidor, **192.168.0.22** y el servicio **5000** correspondiente en NT.
- Debera luego agregar 1 entrada en el Registro de Windows, en **HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/MSSQLServer/**

Client/Connect to como un valor alfanumérico, tal como sigue; **DEVELOPMENT** como nombre del valor y la cadena **dbmssocn,192.168.0.22,5000** como datos del valor.

6.1.9. Creación de una base de datos

En primera instancia se debe crear un ***Database Device*** que servirá para almacenar tanto los datos como el log de la base de datos. Se deberán seguir los siguientes pasos:

1) Abrir el ***Sybase Central***, tal como se muestra en la figura 6.4.

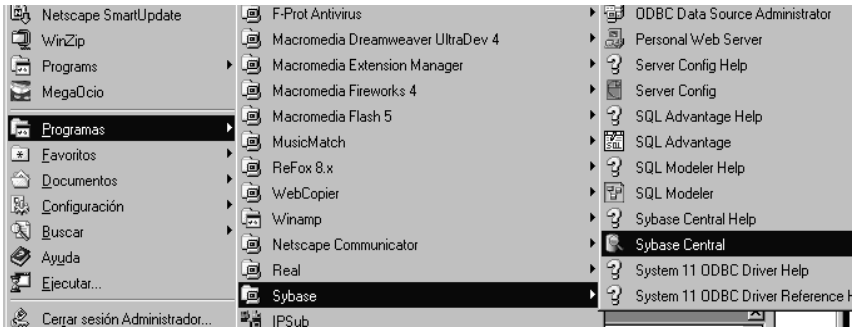


Figura 6.4

2) Debe hacer click con el botón derecho del mouse sobre el servidor de datos (en este caso DEVELOPMENT), espere a que aparezca la ventana de dialogo del login al servidor. La figura 6.5 muestra como debería aparecer.

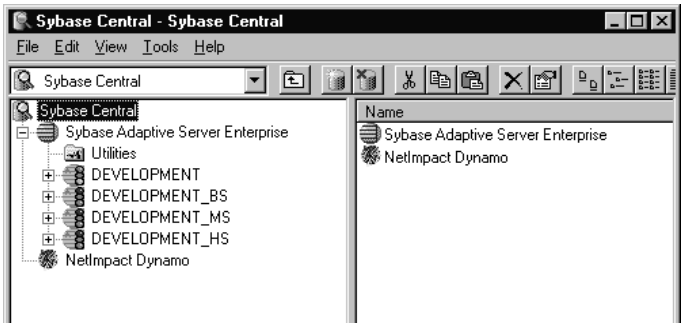


Figura 6.5

3) En esta ventana ingrese el nombre de usuario y la contraseña. Refiérase a la figura 6.6.

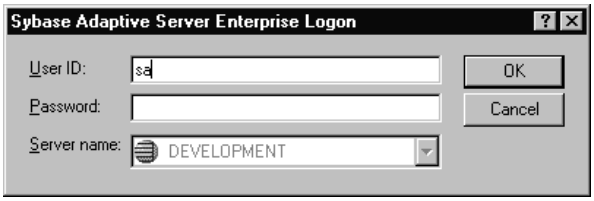


Figura 6.6

4) Ahora ya se puede abrir el servidor de datos, abra la carpeta ***Databases Devices*** y haga click sobre ***Add Database Device*** para agregar un nuevo dispositivo de base de datos, realice este proceso si usted esta seguro que ya no tiene espacio en algún device existente para asignarlo a una base de datos o un log de base de datos. Guíese por la figura 6.7.

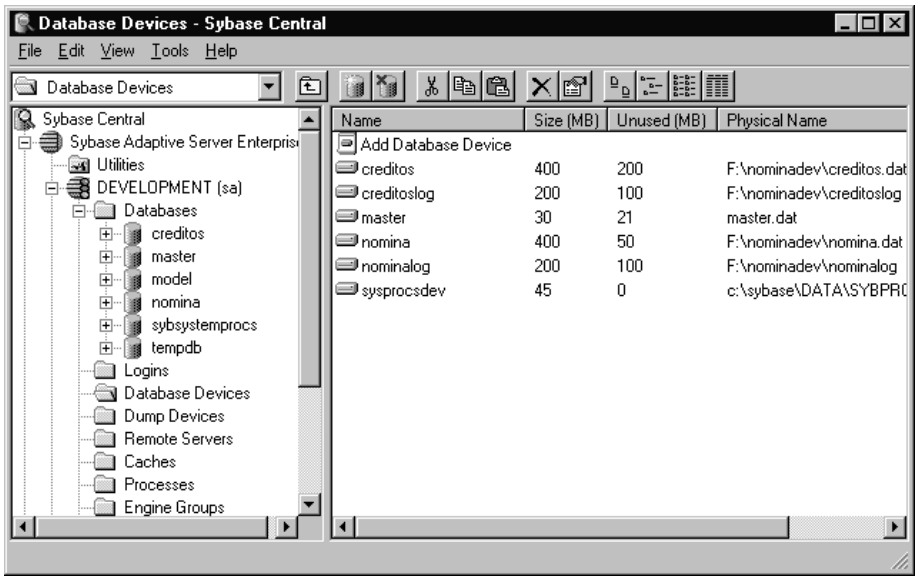


Figura 6.7

5) Ingrese un nombre descriptivo para el device y la ubicación exacta del archivo (.dat) que contendrá el device. En la figura 6.8 se muestra el cuadro de dialogo.

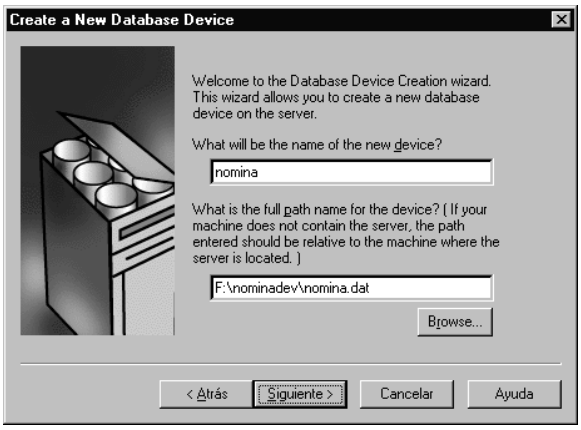


Figura 6.8

6) Ahora debe ingresar el tamaño en Mb del device, no es necesario que cambie el ***Device number***. Esto se muestra en a figura 6.9.

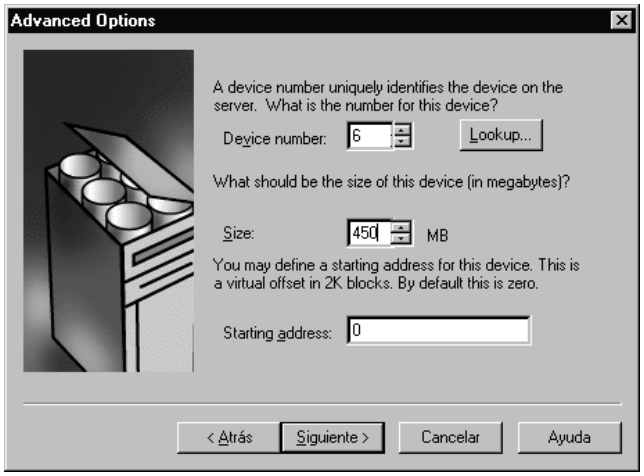


Figura 6.9

7) Si se desea un device espejo para el que se esta creando, se debe especificar aquí (ver figura 6.10), chequeando el control ***Mirror the database device*** e indicar el path completo para el device espejo.

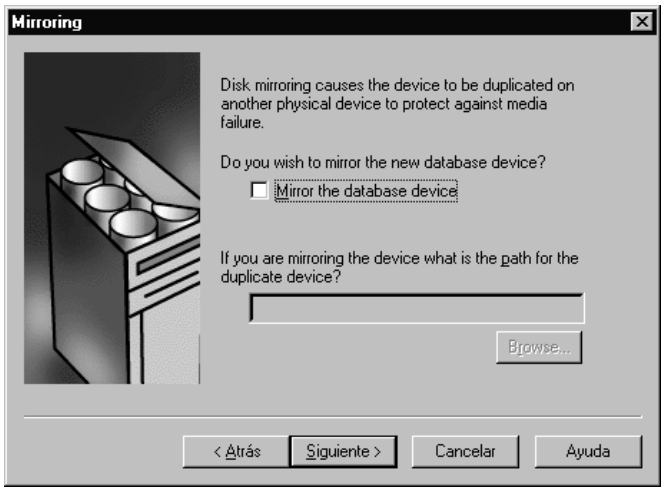


Figura 6.10

8) Con este proceso finalizado aparece el cuadro de dialogo que se muestra en la figura 6.11 y se pasa a la creación física del device, el cual luego estará listo para guardar datos o el log de la base de datos.

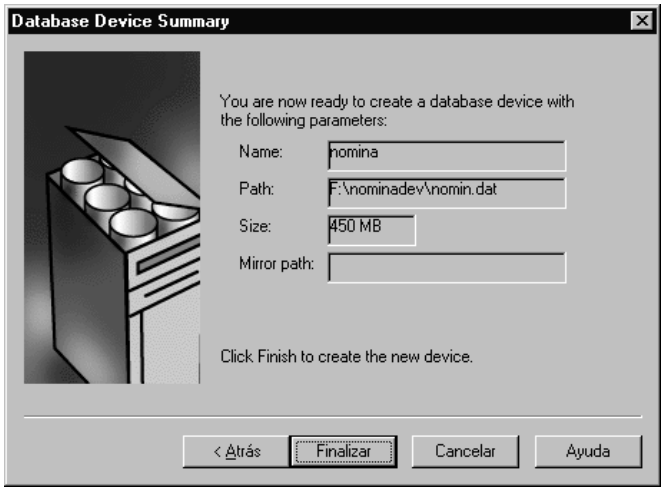


Figura 6.11

9) Ahora se deberá crear la base de datos, haciendo click en ***Add Database*** dentro de la carpeta ***Databases*** en el ***Sybase Central***, tal como lo muestra la figura 6.12.

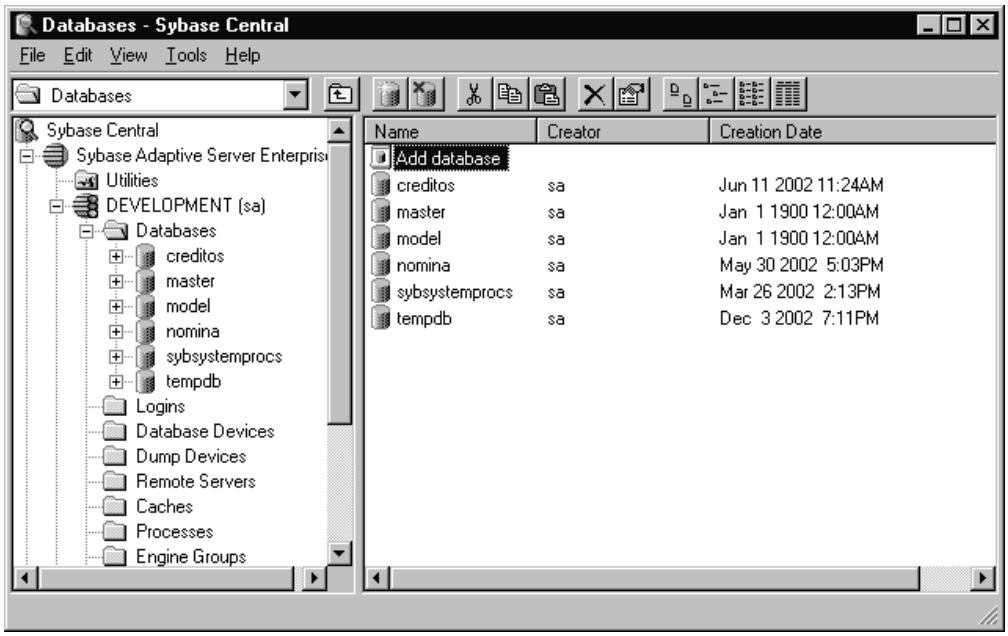


Figura 6.12

10) Ahora deberá ingresar el nombre de la base de datos en el cuadro de dialogo que presenta la figura 6.13.

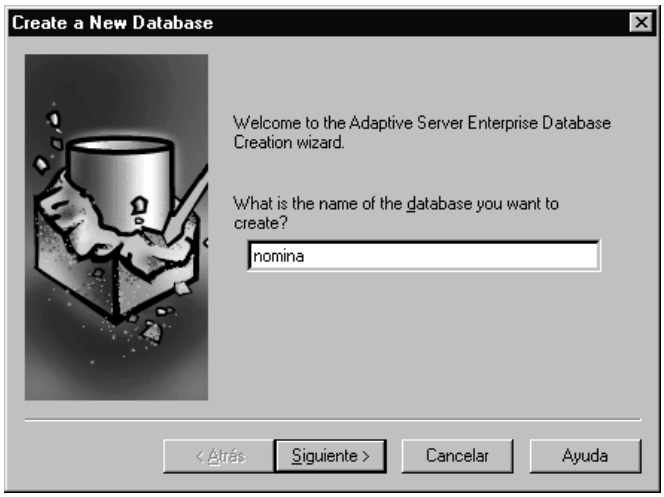


Figura 6.13

11) En este instante deberá escoger un device para la base de datos presionando Add y definir el tamaño de este, además, debe asegurarse de que sea para datos (data), posteriormente deberá realizar el mismo proceso para crear un device para el

transaction log de la base de datos (transaction log).Tal como se muestra en la figura 6.14.

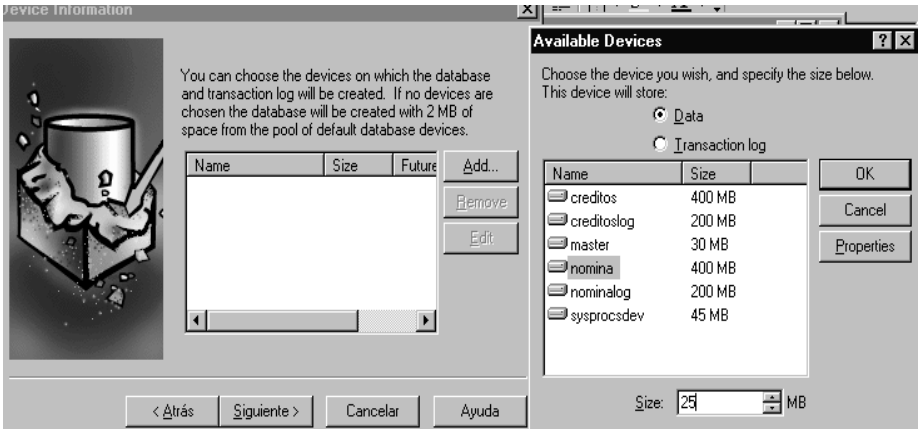


Figura 6.14

12) Ahora ya tiene asignado un device o un conjunto de devices para la base de datos, presione <Siguiente>, tal como se muestra en la figura 6.15

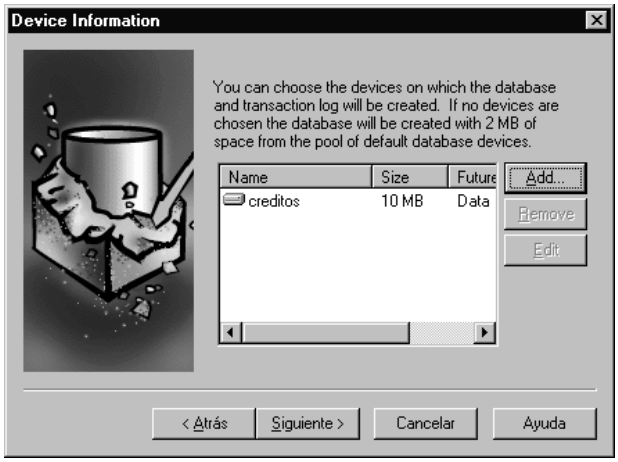


Figura 6.15

13) Si no existe espacio de device suficiente en le servidor, es posible permitir que el log se encuentre en otro device separado, en esta parte se pregunta si el usuario desea que el log siempre se sobrescriba. También se puede indicar que la base de datos sea utilizada para descargar (dump), luego presione <Siguiente>. Mire la figura 6.16.

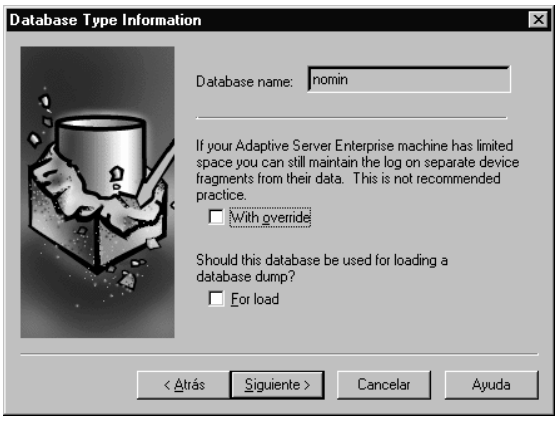


Figura 6.16

14) Si se desea o es necesario, es posible crear un a cuenta de usuario invitado con privilegios limitados solo para esta base de datos. Mire la figura 6.17. Para proceder a la creación de la base de datos, presione <Finalizar>.

Figura 6.17

En este instante ya esta creada la base de datos, además, es posible crearla utilizando sentencias para creación de base de datos con el Transact SQL de ASE Sybase 11.5., de todas formas con algo de experiencia esta seria la manera mas practica de realizar este trabajo. Entonces ahora ya se puede crear otros objetos sobre esta base.

6.1.10. Sybase Central y la administración del servidor

Sybase Central le permite al administrador manejar los objetos en los servidores, las bases de datos (tablas, vistas, procedimientos y mas) y herramientas. También ayuda a realizar tareas administrativas comunes (creación de bases de datos, planificar eventos, instalar consultas dbQ y mas).

Sybase Central simplifica las tareas mencionadas proveyendo una interfase grafica fácil de utilizar, tal como el explorador de Windows 95/98/Me/2000. Por ejemplo, para eliminar una base de datos, se la selecciona en la ventana principal y luego se hace click en <Delete>. Además ayuda en el cumplimiento de tareas mas complicadas proveyendo wizards que le guían al administrador para realizar dicha tarea paso a paso. Sybase Central le permite al administrador utilizar la interfase que mas confortable resulta con menús, barras de tareas, shortcuts o drag'n drop. Además puede administrar una variedad de productos de ases de datos y herramientas utilizando los módulos plug-in

6.1.11. Librerías adicionales

Adicionalmente a las librerías que se instalan por defecto con ***Microsoft Visual Studio 6.0***, se deberá incluir las siguientes referencias para poder editar el proyecto de nomina en ***Microsoft Visual Studio 6.0*** :

Tabla 6.4

Microsoft Visual Basic for Applications
Microsoft ActiveX Data Object 2.5
Microsoft ActiveX Data Object Recordset 2.0
Microsoft Remote Data Object 2.0
Microsoft Remote Data Services 2.6
Microsoft OLE DB Service
Microsoft OLE DB Components 1.0
Microsoft Data Environment Instance 1.0 (SP4)
Microsoft Data Designer 6.0 (SP4)

De igual manera, se debe añadir los siguientes componentes :

Tabla 6.5

Apex Data Bound Grid
Microsoft ADO Data Control 6.0 (SP4) (OLE DB)
Microsoft Data Grid Control 6.0 (SP5)
Microsoft Data List Control 6.0 (SP3)
Microsoft Remote Data Control 6.0 (SP3)
Microsoft Windows Common Controls 5.0 (SP2)

6.1.12. Configuración del Origen de datos ODBC de Sybase System 11

6.1.12.1. Requerimientos del sistema.

Se deberá instalar el **Sybase Open Client Library** (versión 10.03 o mayor para sistemas Intel ®) y la librería apropiada Sybase **Net-Library** para acceder al servidor Sybase.

SYBPING es una herramienta que se provee son Sybase **Net-Library** para probar la conectividad desde una estación cliente hacia el servidos de base de datos (servidores que pueden ser adicionados con ***SQLEdit***). Utilice esta herramienta para probar la conexión.

SQLEdit es una herramienta que permite definir servidores y adicionarlos al archivo **sql.ini** . Se debe establecer la variable de ambiente SYBASE al directorio donde se ha instalado el cliente SYBASE. Se puede establecer esta variable en el Panel de Control bajo Sistema, por ejemplo:

SET SYBASE = C:\ISQL10

6.1.12.2. Creación de un DSN

Para configurar un origen de datos para SYBASE, haga lo siguiente :

a) Inicie el **ODBC Driver Administrator** para desplegar los orígenes de datos.

b) Si se esta configurando un origen de datos existente, seleccione el DSN y haga click en **Configure** para desplegar la ventana de dialogo del ODBC Sybase Driver. Si se esta configurando un nuevo origen de datos. Haga click en **Add** para desplegar la lista de drivers instalados, seleccione **Sybase** y luego haga click en **Finish** para desplegar la ventana de dialogo del **ODBC** Sybase Driver. Refiérase a la figura 6.18.

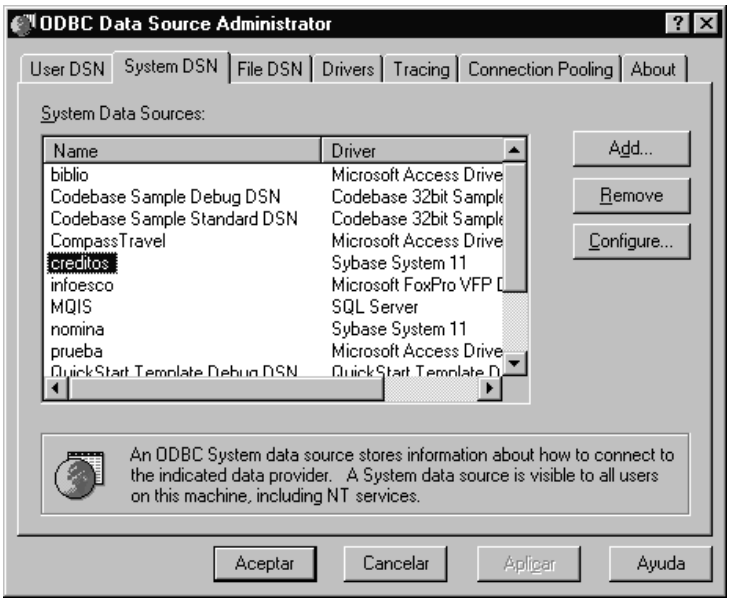


Figura 6.18

c) Especifique un nombre para el origen de datos, un nombre de base de datos y opcionalmente una descripción. Haga click en **Apply** (Aplicar). Ver figura 6.19.

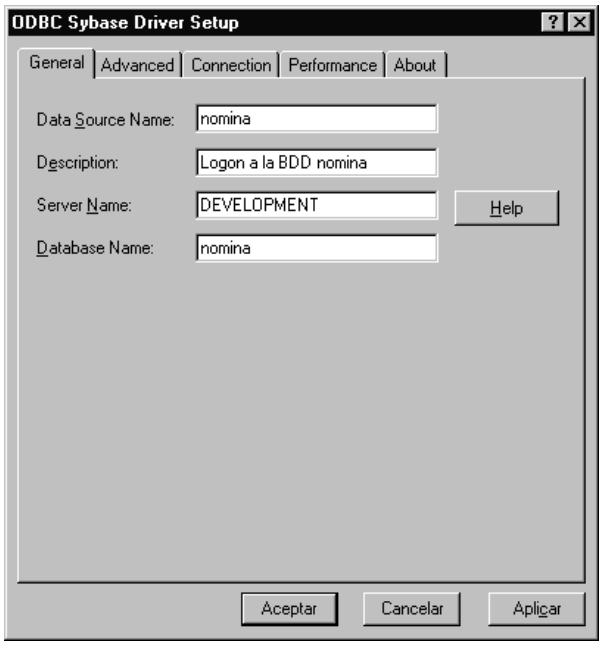


Figura 6.19

d) Haga click en la pestaña **Advanced** para configurar algunos valores opcionales del origen de datos tales como un servidor o un nombre de base de datos, haga un clic en **Apply** (Aplicar). Mire la figura 6.20.

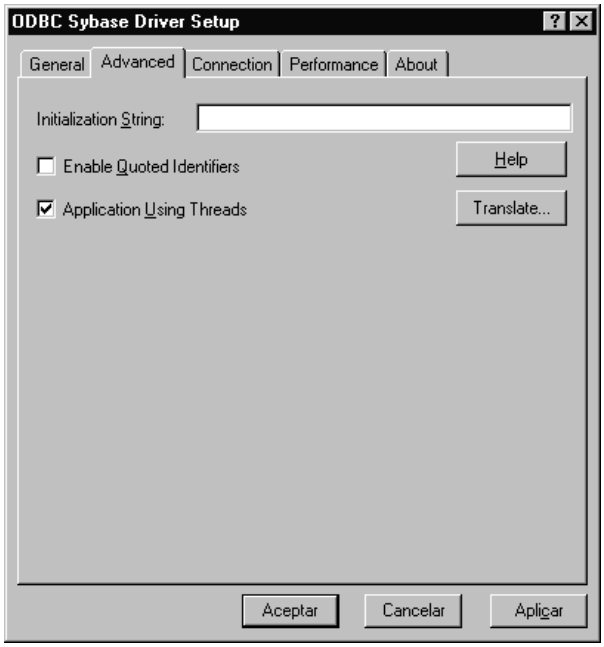


Figura 6.20

e) Haga clic en **Translate**, para configurar la ventana de dialogo **Select Translator** (Ver figura 6.21). Que lista los traductores especificados en la sección **ODBC Translator** del sistema de información. **INTERSOLV** provee un traductor llamado **INTERSOLV OEM ANSI**, que traduce los datos desde un conjunto de caracteres para IBM PC o compatible a un conjunto de caracteres ANSI. Seleccione un traductor y haga un clic en **OK** para cerrar esta ventana de diálogo y realizar la traducción.

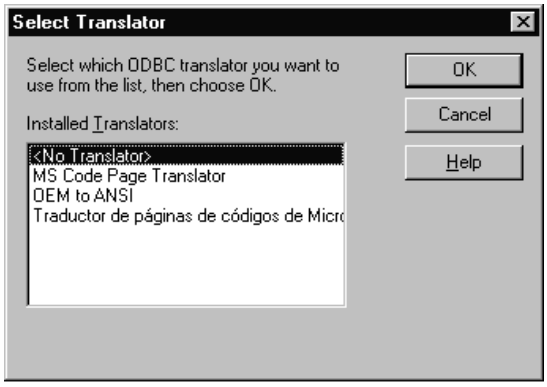


Figura 6.21

f) Haga clic en la pestaña **Connection** para configurar valores opcionales de origen de datos, aparecerá una lista de servidores. Haga clic en **Apply** (Aplicar) como se puede ver en la figura 6.22.

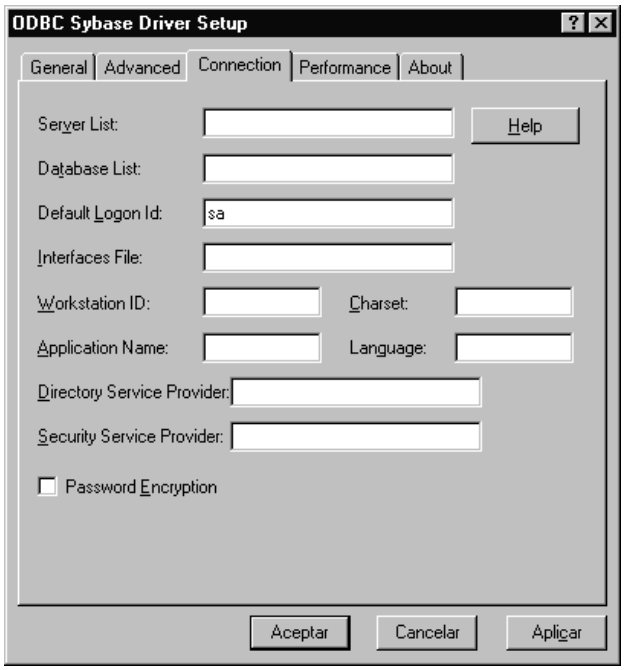


Figura 6.22

g) Haga clic en la pestaña **Performance** (ver la figura 6.23) para configurar opciones avanzadas de rendimiento. Haga clic en **Apply** (Aplicar).

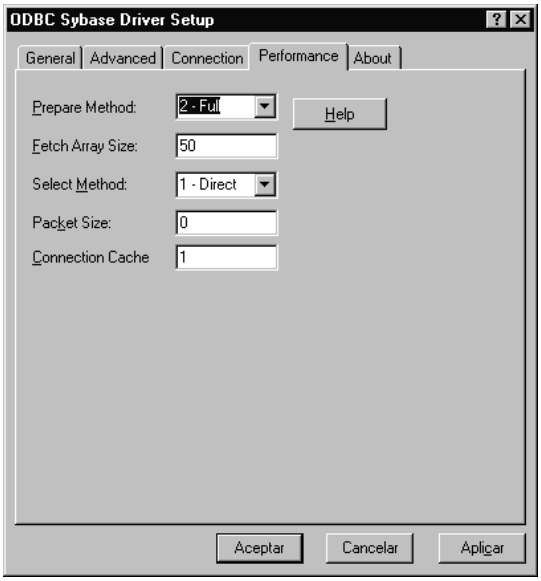


Figura 6.23

h) Haga clic en **OK** o **Cancel**. Si se escogió **OK** los valores que han sido especificados sobrescriben a los valores por defecto cuando se hace la conexión a la fuente de datos. Es posible cambiar estos valores por defecto utilizando este procedimiento para reconfigurar el origen de datos. También puede hacer caso omiso de los valores por defecto conectándose al DSN utilizando una cadena de conexión con valores alternos.

6.1.12.3. Conexión a un origen de datos usando un LOGON DIALOG BOX

Algunas aplicaciones ODBC despliegan un LOGON DIALOG BOX cuando se intenta conectar a un origen de datos. En estos casos, el nombre del origen de datos ya ha sido especificado.

En el LOGON DIALOG BOX, se deberá hacer lo siguiente:

a) Escriba el nombre exacto del servidor que contiene las bases de datos del Sybase System 10 o Sybase System 11 que se desea acceder o seleccione el nombre desde la lista **Server Name**, que despliego los nombres de servidor especificado en el cuadro de diálogo **ODBC Sybase Driver Setup**.
b) Si es requerido escriba el **login ID** exacto.
c) Si es requerido escriba el **password** exacto para el sistema.
d) Escriba el nombre exacto de la base de datos que se desea acceder o seleccione el nombre desde la lista de bases de datos, que despliega los nombres que se han especificado en el cuadro de dialogo **ODBC Sybase Driver Setup**.
e) Haga clic en **OK** para completar el registro y para actualizar los valores en la información del sistema.

6.1.12.4. Conexión a un origen de datos utilizando una cadena de conexión

Si la aplicación requiere una cadena de conexión para conectar a un origen de datos, se debe especificar el nombre del origen de datos que indica al driver que sección en la información del sistema usar para la información de conexión por defecto. Opcionalmente se puede especificar los pares ***atributo = valor*** en la cadena de

conexión para omitir los valores por defecto almacenados en la información del sistema. Estos valores no son escritos en la información del sistema.

Se puede especificar nombres largos o cortos en la cadena de conexión. La cadena de conexión tiene la siguiente forma:

DSN=nombre_origen_datos [;atributo=valor[;atributo=valor]...]

Un ejemplo de una cadena de conexión es:

DNS=MI_DSN;SRVR=SYBASE_SERVER;DB=ROLDEPAGOS;UID=JUAN;PWD=1972

Los párrafos que siguen dan una descripción de los nombres largos o cortos para cada atributo. Los valores por defecto listados son valores por defecto iniciales que aplican cuando no se ha especificado un valor en la cadena de conexión o en la definición del origen de datos en la información del sistema. Si se ha especificado un valor para el atributo cuando se configuró el origen de datos este valor se considerara por defecto.

DataSourceName (DSN): es una cadena que identifica una conexión simple a una base de datos Sybase. Ejemplo: "**cuenta**" o "**mi_dsn**".

ServerName (SRVR): Es el nombre del servidor que contiene tablas de Sybase que se desean acceder. Si no existe valor el valor por defecto inicial es el nombre del servidor en la variable de ambiente **DSQUERY**.

LogonID (UID): el valor por defecto logon ID que es usado para conectar a una base de datos Sybase. Este ID es sensible tanto a mayúsculas como minúsculas. Un logon ID es requerido solamente si la seguridad de la base de datos esta habilitada. En ese caso se debe contactar al administrador del sistema para obtener logon ID.

Database (DB): Es el nombre de la base de datos a la que se quiera conectar.

Languaje (LANG): Es el lenguaje nacional correspondiente a un subdirectorio en **$SYBASE/locales**.

Charset (CS): Es el nombre de un conjunto de caracteres correspondiente a un subdirectorio en **$SYBASE/charsets**.

Con formato

WorkstationID(WKID): Es la ID de una estación de trabajo utilizada por el cliente.

ApplicationName (APP): Es el nombre usado por Sybase para identificar la aplicación.

InterfasesFile (FILE): Es la ruta completa y el nombre del archivo de interfases.

ArraySize (AS): Es el número de filas que el driver recupera desde el servidor por cada entrega. Este no es el número de filas dados por el usuario. Incrementa el rendimiento reduciendo el trafico de la red. El valor inicial por defecto son 10 filas.

OptimizePrepare (OP): **OptimizePrepare={0|1|2}**. Es un valor que determina si los procedimientos almacenados son creados sobre el servidor para cada llamada a **SQLPrepare**. Cuando este atributo se establece a cero, los procedimientos almacenados son creados para cada llamada a SQL Prepare. Este valor puede dar como resultado un pobre rendimiento cuando se esta procesando sentencias estáticas. Cuando se establece uno el valor inicial por defecto, el driver crea procedimientos almacenados solamente si la sentencia contiene parámetros. Por otra parte, la sentencia es recogida y ejecutada directamente con **SQLExecute**. Cuando se establece a 2, el driver nunca crea procedimientos almacenados. Ese tributo es ignorado para servidores Sybase 4.9.9.

SelectMethod (SM): **SelectMethod ={0|1}**. Es un valor que determina si los cursores de las bases de datos son utilizados por sentencias Select. Cuando se establece a 0, valor por defecto inicial, los cursores de base de datos son usados. En algunos casos puede ocurrir que el rendimiento se degrade cuando se realiza un gran numero de sentencias Select secuenciales, porque crece la cantidad de sobrecarga asociada con la creación de cursores de base de datos. Cuando se establece a 1, las sentencias Select son ejecutadas directamente sin utilizar cursores de base de datos.,

cuando se establece a 1, el origen de datos está limitado a una sentencia activa y a una conexión activa. Este atributo es ignorado para servidores Sybase 4.9.2.

PasswordEncyption (PE): PasswordEncyption={-1|0|x}. Es un dígito que determina el número de bytes por paquete de red transferido desde el servidor de base de datos hacia el cliente. El correcto establecimiento de este atributo puede mejorar el rendimiento. Cuando se establece a 0, el valor por defecto inicial, el driver utiliza el tamaño del paquete por defecto tal como esta especificado en la configuración del servidor Sybase. Cuando se establece el atributo a –1, el driver calcula el tamaño máximo permitido del paquete en la primera conexión al origen de datos y guarda este valores en la información del sistema. Cuando se estable el parámetro a **x**, donde **x** es un entero entre 1 y 10, indica el múltiplo de 512 bytes (por ejemplo, PacketSize = 6 significa colocar el tamaño del paquete a 6*512 = 3072 bytes). Para tomar ventaja de este atributo de conexión de debe configurar el servidor Sybase para el máximo tamaño del paquete de la red mayor o igual al valor especificado por PacketSize; por ejemplo:

Sp_configure "maximum network packet size", 5120
Reconfigure
Restart Sybase Server

Note que la especificación ODBC indica una opción de conexión, **SQL_PACKET_SIZE**, que ofrece la misma funcionalidad. Para evitar conflictos con aplicaciones que pueden establecer ambos atributos de conexión, es decir la cadena de conexión y la opción de conexión ODBC, estos han sido definidos como mutuamente excluyentes. Si **PacketSize** es especificado se recibirá un mensaje tal como "Driver Not Capable" si se intenta llamar a **SQL_PACKET_SIZE**. Si no es establece **PacketSize**, entonces la aplicación llama a **SQL_PACKET_SIZE** que es aceptado por el driver.

CursorCachéSize (CCS): Es un valor que determina el numero de conexiones que el caché de conexión puede albergar. El valor por defecto para CursorCachéSize es 1, para establecer el caché de conexión , se debe establecer la opción SelectMethod a 1. Al incrementar el caché de conexión se puede incrementar el rendimiento de algunas aplicaciones pero requiere recursos adicionales de la base de datos.

ApplicationUsingThreads (**AUT**): **ApplicationUsingThreads={0|1}**. Asegura que el driver trabaje con aplicaciones multihebra. El valor por defecto es 1, que hace l driver **Thread – Safe**. Cuando se utiliza el driver con aplicaciones de hebra simple, se puede colocar esta opción a cero para evitar procesamiento adicional requerido por los estándares de seguridad del **ODBC Thread**.

EnableQuotedIdentifiers (EQI): EnableQuotedIdentifiers = {0|1}. Especifica uno para permitir soporte de identificadores marcados. El valor por defecto es 0. Este atributo esta disponible para servidores System10 y System11.

InitializationString (IS)= InitializationString={<Sybase set command>;...}. Soporta la ejecución de comandos Sybase en tiempo de conexión. Su existen múltiples comandos deben estar separados por puntos y comas.

DirectoryServiceProvider (DSP): Es una cadena que indica cual DSP utiliza el Sybase Open Client cuando se conecta con un origen de datos. El DSP disponible puede ser encontrado utilizando la herramienta **OpenClient/OpenServer Configuration Utility** que esta instalada con **Sybase Open Client versión 11.1** o mayor. Si el cliente no esta utilizando **Sybase Open Client versión 11.1** o mayor esta opción es ignorada.

SecurityServiceProvider (SSP): Es una cadena que indica cual SSP utiliza el Sybase Open Client cuando se conecta con este origen de datos . El SSP disponible puede ser encontrado utilizando la herramienta **OpenClient/OpenServer Configuration Utility** que esta instalada con **Sybase Open Client versión 11.1** o mayor. Si el cliente no esta utilizando **Sybase Open Client versión 11.1** o mayor esta opción es ignorada..

6.1.12.5. Tipos de datos soportados

Los tipos de datos de Sybase son trazados por los tipos de datos estándar del ODBC tal como sigue:

Tabla 6.6

SYBASE	ODBC
Binary	SQL_BINARY
Bit	SQL_BIT
Char	SQL_CHAR
Datetime	SQL_TYPE_TIMESTAMP
* decimal	SQL_DECIMAL
Float	SQL_FLOAT
Image	SQL_LONGVARBINARY
Int	SQL_INTEGER
Money	SQL_DECIMAL
* numeric	SQL_NUMERIC
Real	SQL_REAL
Smalldatetime	SQL_TYPE_TIMESTAMP
Smallint	SQL_SMALLINT
Smallmoney	SQL_DECIMAL
Sysname	SQL_VARCHAR
Text	SQL_LONGVARCHAR
Timestamp	SQL_VARBINARY
Tinyint	SQL_TINYINT
Varbinary	SQL_VARBINARY
Varchar	SQL_VARCHAR

* No soportado por Sybase 4.9.9

6.1.12.6. Niveles soportados de aislamiento y cerrojo

Sybase soporta niveles de aislamiento (**isolation**) 0 (si la versión de servidor es 11 o mayor), 1 (por defecto read committed), y 3 (serializable). Este soporta cerrojos (**look**) a nivel de página.

6.1.12.7. Nivel de conformidad ODBC

Las funciones API soportadas están listadas en: **"Supported ODBC Functions"**, que se encuentran en la **General Help** del **DataDriver ODBC Drivers**, adicionalmente las siguientes funciones de nivel 2 son soportadas.

SQLColumnPrivileges
SQLForeignKeys
SQLPrimaryKeys

SQLProcedureColumn
SQLProcedures
SQLTablePrivileges

El driver de Sybase soporta la gramática mínima de SQL.

6.1.12.8. Número de conexiones y sentencias soportadas

EL Sybase Database System soporta múltiples conexiones y múltiples sentencias por conexión.

6.1.12.9. Ventana de diálogo ODBC SYBASE DRIVER SETUP

Utilice la ventana ***ODBC SYBASE DRIVER SETUP*** para crear un nuevo origen de datos Sybase o configurar un origen de datos existente.

- General Tab

Data Source Name: Es una cadena que identifica la configuración de un origen de datos Sybase, dentro de la información del sistema. Un ejemplo puede ser: "Créditos" o "Tesorería".

Description: Es una descripción larga opcional de un nombre de origen de datos. Por ejemplo: "Créditos vencidos del año anterior" o "Información anual de tesorería".

DataBase Name: Es el nombre de la base de datos a la cual se desea conectar por defecto. Si no especifica un valor, el valor por defecto es la base de datos definida para cada usuario por el administrador del sistema.

Server Name: Es el nombre del servidor que contiene las tablas Sybase que se desea acceder. Si no se indica este nombre, se utiliza el nombre del servidor guardado en la variable de ambiente **DSQUERY.**

- Advanced Tab

Despliega algunos datos opcionales a configurar para el origen de datos, por ejemplo la cadena de inicialización

Utilice la ventana de dialogo Advanced Tab para establecer valores opcionales cuando se crea un nuevo origen de datos o configura una base de datos existente.

Initialization String: soporta la ejecución de comandos Sybase en tiempo de conexión cuando existen múltiples comandos, estos deber ser separados por puntos y comas.

Enable Quoted Identifiers: Es una configuración que permite el soporte de identificadores marcados en servidores System 10 o System 11.

Application Using Threads: especifica un valor que asegura que el driver trabaje con aplicaciones multihebras. Se puede limpiar este Check Box cuando se esta utilizando el driver de una sola hebra. Apagando esta opción se evita procesamiento adicional requeridos por el estándar de seguridad de hebras ODBC.

Translate Button: despliega el cuadro de Select Translator, donde se puede traducir los datos de un conjunto de caracteres a otro. Se puede escoger el traductor INERSOLV OEM ANSI, para traducir los datos desde un conjunto de caracteres IBM PC o compatible a un conjunto de caracteres ANSI.

- Connection Tab

Utilice el Connetion Tab en la ventana ODBC Sybase Driver Setup, para especificar valores de configuración opcionales, cuando se crea un nuevo origen de datos o se configura un existente.

Server List: Es la lista de servidores que aparecen en el cuadro de dialogo Logon. Se indican los nombres de los servidores separados por comas.

Data Base List: Son las bases de datos que aparecen en el cuadro de dialogo Logon. Se indican los nombre de bases de datos separados por comas.

Default Logon ID: Es el logon ID por defecto utilizado para conectar a una base de datos Sybase. Este ID es sensible a caracteres en minúscula o mayúscula. El Logon ID es requerido solamente si la seguridad de la base de datos está habilitada. Una aplicación ODBC puede obviar este valor o el usuario puede obviar este valor en el cuadro de dialogo Logon o en la cadena de conexión.

Interfases File: Es la ruta y nombre completo del archivo de interfases. Por defecto es el archivo de interfases normal de Sybase.

Workstation ID: Es el ID de una estación de trabajo utilizado por el cliente.

Charset: Es el nombre de un conjunto de caracteres correspondiente a un subdirectorio en ***$SYBASE/charset***. Es el valor por defecto esta en el servidor Sybase.

Application Name: Es el nombre utilizado por Sybase para identificar una aplicación.

Languaje: Es el lenguaje nacional correspondiente a un subdirectorio en ***$SYBASE/locales***. El valor por defecto es English.

DirectoryServiceProvider (DSP): Es una cadena que indica cual DSP utiliza el Sybase Open Client cuando se conecta con este origen de datos. El DSP disponible puede ser encontrado utilizando la herramienta OpenClient/OpenServer Configuration Utility que esta instalada con Sybase Open Client versión 11.1 o mayor. Si el cliente no esta utilizando Sybase Open Client versión 11.1 o mayor esta opción es ignorada.

SecurityServiceProvider (SSP): Es una cadena que indica cual SSP utiliza el Sybase Open Client cuando se conecta con este origen de datos . El SSP disponible puede ser encontrado utilizando la herramienta OpenClient/OpenServer Configuration Utility que esta instalada con Sybase Open Client versión 11.1 o mayor. Si el cliente no esta utilizando Sybase Open Client versión 11.1 o mayor esta opción es ignorada.

Password Encryption: Es un numero que determina si puede ser realizada la encriptación del password desde la Open Client Library al servidor (PasswordEncryption=1).

Cuando se establece este valor a 0, por defecto no se realiza la encriptación.

- Perfomance Tab

Utilice el **Perfomance Tab** para especificar valores opcionales de configuración cuando se crea un nuevo origen de datos Sybase o se configura uno existente.

Prepared Method: Es un valor que determina si los procedimientos almacenados son creados en el servidor para cada llamada a **SQLprepared**. Cuando se establece a cero los procedimientos almacenados son llamados a **SQLprepared**, que pueden degradar el rendimiento cuando se esta procesando sentencias estáticas. Cuando se establezca uno, el valor inicial por defecto el driver crea los procedimientos almacenados solamente si la sentencia contienen parámetros de otra forma la sentencia es atrapada y ejecutada directamente cuando se llama a **SQLExecute**. Cuando se establece el valor a 2 el driver no crean procedimientos almacenados. Este valores de configuración es ignorado cuando se intenta conectar a servidores Sybase versión 4.9.2 .

Fetch Array Size: Es el número de filas que el driver recupera desde el servidor por cada entrega. Este no es el número de filas dados por el usuario. Incrementa el rendimiento reduciendo el trafico de la red. El valor inicial por defecto son 10 filas.

Select Method : Es un valor que determina si los cursores de las bases de datos son utilizados por sentencias Select. Cuando se establece a 0, valor por defecto inicial, los cursores de base de datos son usados. Cuando se establece a 1, las sentencias Select son ejecutadas directamente sin utilizar cursores de base de datos., cuando se establece a 1, el origen de datos está limitado a una sentencia activa y a una conexión activa. Este atributo es ignorado para servidores Sybase 4.9.2.

Packet Size : Es un valor que determina el numero de bytes por paquete de red transferido desde el servidor de base de datos hasta el cliente. El correcto establecimiento de este atributo puede mejorar el rendimiento. Cuando el valor se establece a 0, el valor por defecto inicial, el driver utiliza el tamaño del paquete por defecto como esta especificado en la configuración del servidor Sybase. Cuando se establece a –1, el driver calcula el máximo tamaño de paquete permitido en la primera conexión al origen de datos y guarda el valor en la información del sistema. Cuando se estable el parámetro a **x**, donde **x** es un entero entre 1 y 10, indica el múltiplo de 512 bytes (por ejemplo, PacketSize = 6 significa colocar el tamaño del paquete a 6*512 = 3072 bytes). Para tomar ventaja de este atributo de conexión de debe configurar el servidor Sybase para el máximo tamaño del paquete de la red mayor o igual al valor especificado por PacketSize; por ejemplo:

sp_configure "maximum network packet size", 5120
Reconfigure
Restart Sybase Server

Note que la especificación ODBC indica una opción de conexión, SQL_PACKET_SIZE, que ofrece la misma funcionalidad. Para evitar conflictos con aplicaciones que pueden establecer ambos atributos de conexión, es decir la cadena de conexión y la opción de conexión ODBC, estos han sido definidos como mutuamente excluyentes. Si PacketSize es especificado se recibirá un mensaje tal con "Driver Not Capable" si se intenta llamar a SQL_PACKET_SIZE. Si no es establece PacketSize, entonces la aplicación llama a SQL_PACKET_SIZE que es aceptado por el driver.

Connection Caché : Es un valor que determina el numero de conexiones que el caché de conexión puede albergar. El valor por defecto para el caché de conexión es 1, para establecer el caché de conexión , se debe establecer la opción SelectMethod a 1 (Direct). Al incrementar el caché de conexión se puede incrementar el rendimiento de algunas aplicaciones pero requiere recursos adicionales de la base de datos.

6.1.13.10 . Logon a la ventana de diálogo SYBASE

La ventana de dialogo LOGON de Sybase contiene los siguientes requerimientos:

Server Name : Aquí se debe ingresar el nombre exacto del servidor que contiene las bases de datos que se intenta acceder, o seleccionar uno de la lista desplegable, los nombres de los servidores deben haber sido especificados en la ventana de dialogo ODBC Sybase Driver Setup.

Login ID : Si acaso es requerido, ingrese el login ID exacto.

Password : Si acaso es requerido, ingrese su password exacto para el sistema.

Database : Ingrese el nombre exacto de la base de datos que se desea agregar o seleccione uno de la lista desplegable, los nombres de bases de datos deben haber sido especificados en la ventana de dialogo ODBC Sybase Driver Setup, vea la figura 6.24.

Figura 6.24

6.1.13.11. Sybase System 11 ODBC Reference Help

Esta ayuda contiene información referente para el Sybase System 11 ODBC.

- About Sybase ODBC drivers
- For Help on a Database Driver
- Environment - Specific Information
- ODBC functions Supported
- Error Messages

6.1.13.12. Acerca de Sybase System ODBC driver

El driver ODBC de Sybase System 11 cumple con la especificación ODBC, esta es una especificación para la interfaz de programas de aplicación (API) que habilita a las aplicaciones para que puedan acceder a múltiples DBMS's utilizando SQL.
El estándar ODBC permite una máxima interoperatibilidad: una simple aplicación puede acceder diferentes DBMS's . Esto permite a un desarrollador ODBC, crear, compilar y enviar una aplicación sin indicar un origen de datos especifico como destino.. Los usuarios además pueden guardar los drivers de base de datos, que enlazan la aplicación al DBMS que ellos escojan.

6.1.13.13. Utilización de drivers ODBC sobre Windows 95/98/Me/NT/2000

En los sistemas Windows los drivers trabajan a 32 bits. Todo el software de red proporcionado por los vendedores de bases de datos deben trabajar por ende a 32 bits. Para conocer los requerimientos específicos de cada driver de bases de datos relacionales, haga click sobre **Contents** en el submenú **Help** de la barra de herramientas de Windows, luego haga click sobre la pestaña **Contents** en la ventana **Help Topics** . Ahora haga click sobre el icono del libro de cualquier driver, y vuelva a hacer click sobre el tópico **"System Requeriments" .**

6.1.13.14. El ODBC Data Source Administrator desde el punto de vista de Sybase

Nombres de Drivers .- En Windows NT y Windows 95, todos los drivers ODBC para Sybase inician con las letras **SY** y finalizan con **NT** o **95** según sea el caso, la extensión del archivo es **.dll**, por ejemplo, el driver ODBC de Sybase sobre una plataforma **nn** seria **SYSYBnn.DLL.** Para conocer los nombres de archivos específicos de cada driver de bases de datos relacionales, haga click sobre **Contents** en el submenú **Help** de la barra de herramientas de Windows, luego haga click sobre la pestaña **Contents** en la ventana **Help Topics** . Ahora haga click sobre el icono del libro de cualquier driver, y vuelva a hacer click sobre el tópico **"About the driver" .**

***Requerimientos de espacio de disco y memoria*.-** Se requiere 6 Mb. de espacio de disco libre sobre cualquier plataforma Windows. El requerimiento de memoria varia dependiendo del driver de base de datos. Si se utiliza un driver ODBC flat-file por lo menos se requiere 8 Mb de memoria sobre Windows 95/98/Me o al menos 16 Mb de memoria sobre Windows NT/2000. Se debe consultar directamente la documentación de cada DBMS para determinar los requerimientos exactos de memoria.

6.1.13.15. Soporte de funciones ODBC

Los drivers ODBC DataDirect soportan las funciones API indicadas en la especificación Microsoft ODBC. En general, todos los drivers ODBC soportan :

- ODBC 2.x Conformance Functions
- ODBC 3x Conformance Functions

Algunas drivers soportan funciones adicionales o soportan funciones especificas de manera diferente. Estas diferencias están listadas en el tópico "ODBC Conformance Level" de cada driver.

6.1.13.16. Descripción de Objetos y controles ADO

Microsoft ActiveX Data Objects (ADO) es una interfaz basada en Automatización para obtener acceso a datos. ADO utiliza la interfaz de OLE DB para tener acceso a un amplio conjunto de orígenes de datos, incluyendo pero no limitándose a los datos proporcionados mediante ODBC. Los usuarios de RDO y DAO deberán sentirse rápidamente cómodos con la programación en ADO, ya que el diseño general de ADO proviene del desarrollo de estas interfaces.

En la actualidad ADO viene conjuntamente con el ***Option Pack*** de Windows NT 4, dentro del Option Pack están los instaladores para Windows 95/98/Me, Windows NT WorkStation, Windows NT Server.

La nueva versión 6.0 de Visual Basic trabajará con ADO naturalmente, por eso es importante comenzar todos los nuevos proyectos con esta interfaz. También la versión de SQL Server 7.0 puede trabajar con ADO, lo que le posibilita al motor de base de datos conectarse a cualquier proveedor OLE DB, por ejemplo, desde un procedimiento

almacenado de SQL Server se puede acceder a una base de datos Access para consultar datos, o realizar una búsqueda con Microsoft Index Server (indexador de todo tipo de documentos Word, Excel, etc.) y encontrar documento por una determinada cadena de caracteres.

Entre las características de los objetos ADO son de mención especial las siguientes:

- Cursores del lado del cliente, como RDO con actualización optimista, objetos Recordset sin conexión y mucho más.
- Comandos como métodos de conexión, los comandos asociados con esta conexión se convierten en métodos.
- Acceso a datos remotos, los usuarios de ADO pueden transmitir datos a través de HTTP a un cliente, trabajar con dichos datos devolverlos al servidor HTTP de nuevo.

Para conectarse por medio de ADO es necesario un proveedor de OLE DB, en el Option Pack están disponibles tres proveedores:

a) *OLE DB Provider for ODBC*: Permite conectarse a cualquier fuente de datos ODBC. Los drivers ODBC están disponibles para la mayoría de los DBMS en uso hoy.

b) *OLE DB Provider for Microsoft Index Server*: El proveedor para Microsoft Index Server proporciona acceso de solo lectura a los archivos de sistema y documentos Word, Excel, Web, etc. Las consultas SQL pueden recibir la información en forma apropiada para el motor de base de datos.

c) *OLE DB Provider for Microsoft Active Directory Service*: Permite acceder a los servicios de directorios, tanto de NT como de Novell.

El siguiente ejemplo muestra las diferentes maneras de usar la propiedad **ConnectionString** para abrir un objeto **Connection**. También usa la propiedad de **ConnectionTimeout** para poner un periodo de interrupción de conexión, y la propiedad **state** para verificar el estado de las conexiones.

```
Public Sub ConnectionString()

        Dim cnn1 As ADODB.Connection
        Dim cnn2 As ADODB.Connection
        Dim cnn3 As ADODB.Connection
        Dim cnn4 As ADODB.Connection

        ' Abre una conexión sin usar un Data Source Name (DSN).
        Set cnn1 = New ADODB.Connection
        cnn1.ConnectionString = "driver={SQL Server};" & _
                "server=bigsmile;uid=sa;pwd=pwd;database=pubs"
        cnn1.ConnectionTimeout = 30
        cnn1.Open

        ' Abre una conexión usando DSN y ODBC.
        Set cnn2 = New ADODB.Connection
        cnn2.ConnectionString = "DSN=Pubs;UID=sa;PWD=pwd;"
        cnn2.Open

        ' Abre una conexión usando DSN y OLE DB.
        Set cnn3 = New ADODB.Connection
        cnn3.ConnectionString = "Data Source=Pubs;User
        ID=sa;Password=pwd;"
        cnn3.Open

        ' Abre una conexión que usa un DSN y argumentos
        ' individuales en lugar de una cadena de conexión.
        Set cnn4 = New ADODB.Connection
        cnn4.Open "Pubs", "sa", "pwd"

        ' Muestra el estado de las conexiones
        MsgBox "cnn1 state: " & GetState(cnn1.State) & vbCr & _
                "cnn2 state: " & GetState(cnn2.State) & vbCr & _
                "cnn3 state: " & GetState(cnn3.State) & vbCr & _
                "cnn4 state: " & GetState(cnn4.State)

        cnn4.Close
        cnn3.Close
        cnn2.Close
```

```
        cnn1.Close

End Sub

Public Function GetState(intState As Integer) As String

        Select Case intState
                Case adStateClosed
                        GetState = "adStateClosed"
                Case adStateOpen
                        GetState = "adStateOpen"
        End Select

End Function
```

ADO contiene la colección de objetos para crear una conexión a bases de datos y leer datos desde tablas, trabajando como una interfaz hacia la fuente de datos (OLE DB).

En general, después de crear una conexión a la base de datos, se puede ignorar la existencia de OLE DB, debido a que este driver hace todo su trabajo en "background".

Existen dos maneras para que el proveedor OLE DB brinde acceso a una base de datos:

a) *Indirectamente*
De este manera se accede mediante un driver ODBC**.**

b) *Directamente*
De esta manera se accede mediante un driver OLE DB nativo**.**

Se debe destacar también los métodos para acceder a bases de datos Microsoft SQL Server.

a) *Conectar a SQL Server con OLE DB*

El modo preferido para conectar a una base de datos SQL Server, es utilizar un proveedor nativo OLE DB para SQL Server. Este ejemplo abre una conexión a la base de datos **Pubs** en un servidor Microsoft SQL Server llamado **tuBase.**

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "PROVIDER=SQLOLEDB;DATA SOURCE=tuBase;
UID=sa;PWD=secret;DATABASE=Pubs
%>
```

Este script crea una instancia del objeto **Connection**. Luego la conexión a SQL Server es abierta con un script de conexión que contiene 5 parámetros:

- El proveedor OLE DB (Provider),
- La fuente de datos (data source),
- La base de datos (database),
- El nombre de usuario (user ID)
- La clave (password).

El parámetro PROVIDER es utilizado para especificar el nombre del proveedor OLE DB a usar en la conexión, este ejemplo utiliza un proveedor OLE DB nativo para SQL Server. En es caso de no especificar ningún proveedor, se utilizará el driver ODBC para OLE DB por defecto. El parámetro DATA SOURCE, es utilizado para proveer el nombre del Servidor SQL. El parámetro UID indica el login para acceder al servidor SQL Server. Este ejemplo utiliza el login ***sa***, sin embargo, es conveniente utilizar otro login, por cuestiones de seguridad. Igualmente el parámetro UID es opcional, si no se coloca nada en el, la conexión será hecha con la cuenta ***IUSER_MachineName***. El parámetro PWD contiene la clave para acceder al servidor SQL. Finalmente el parámetro DATABASE es utilizado para especificar una particular base de datos, localizada en el Servidor SQL.

b) *Conectar a SQL Server con ODBC*

El método tradicional para conectar una base de datos es a través de ODBC, y existen tres maneras para hacer esto:

- Guardando la información en el registro de Windows
- En un archivo de texto
- Dentro del string de conexión mismo.

Antes de crear el script para almacenar la información de la conexión es necesario crear un DNS de Sistema en el registro de Windows. Se debe ingresar un nombre y una descripción para el DSN, y después seleccionar el **Servidor o Server,** al cual se necesita conectar. Si el SQL Server esta localizado sobre alguna máquina con IIS (Internet Información Server), hay que seleccionar **Local.** Si en cambio, el Servidor esta localizado sobre Internet, hay que ingresar una dirección **IP** o un **nombre de dominio o domain name.**

En lugar de almacenar la información de la conexión en el ***Windows Registry,*** como se indica en el párrafo anterior, es posible guardar esta información en un archivo de texto, creando un **File DNS o DNS de Archivo.** Para crear este archivo, hay que como siempre, abrir la fuente de datos ODBC, seleccionar **File DNS o DNS de Archivo**, presionar **Agregar o Add** y especificar el nombre y un lugar donde será almacenado el archivo que contendrá la información de la conexión.

Si por ejemplo, el DNS de Archivo que se creó, de llamara **MiArchivoDNS,** se tendría que utilizar un string como este para abrir la conexión a la base de datos.

```
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open = "FILEDNS=MiArchivoDNS;UID=sa;PWD=secret;DATABASE=Pubs"
```

También se puede colocar la información de la conexión dentro del script. Este método es sumamente sencillo, y no es necesario configurar nada en el Administrador ODBC.

```
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DRIVER={SQL Server};SERVER=NombreServidor;UID=sa;PWD=secret"
```

Todos estos métodos son sencillos de implementar y la elección de alguno de ellos dependerá del tipo de conexión, seguridad e información que se tenga del servidor.

Se debe mencionar además que es posible adicionar a un formulario de Visual Basic un control visible ADO (ADOC), la ventaja de este control radica en que es posible configurar y conocer los orígenes de datos en tiempo de diseño, además existen otros controles orientados a datos que necesariamente deben interactuar con un ADOC. La ventaja del código ADO radica en cambio en la versatilidad que tiene el programador para navegar y reorientar el objetivo de la codificación.

6.1.13.17. Código y controles de conexión del prototipo

La conexión a la base de datos del prototipo se realiza de 3 formas diferentes pero basadas en un mismo origen de datos.

La primera forma es la conexión por medio de código de ADO, en la figura NN se muestra un formulario para ingreso de nuevos periodos de cálculo de la nomina de pago, en la parte inferior del mismo se observa (aparentemente) un control de acceso a datos, en realidad son controles básicos de Visual Basic que emularan el comportamiento de un control de datos ya que el proceso de conexión a datos se realiza mediante código de ADO interno, el mismo se detalla luego de la figura 6.25.

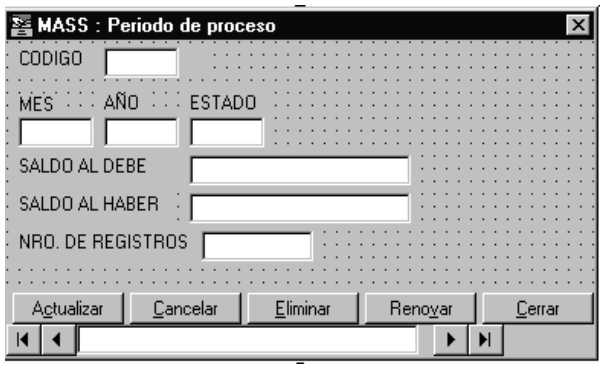


Figura 6.25

```
Option Explicit
Dim WithEvents adoPrimaryRS As Recordset

Private Sub Form_Load()
  Dim db As Connection
  Set db = New Connection
  db.CursorLocation = adUseClient
```

```
  db.Open "PROVIDER=MSDASQL;dsn=nomina;uid=sa;pwd=;database=nomina;"
  Set adoPrimaryRS = New Recordset
  adoPrimaryRS.Open & _
   "select PER_COD,PER_MES,PER_ANO,PER_ESTADO,PER_SAL_DEBE," & _
   "PER_SAL_HABER,PER_NRO_REGS from TPERIODO", & _
   db, adOpenStatic, adLockOptimistic

Dim oText As TextBox
  'Enlaza los cuadros de texto con el proveedor de datos
  For Each oText In Me.txtfields
    Set oText.DataSource = adoPrimaryRS
  Next
End Sub
```

Se declara en primera instancia una variable Recordset de nombre ***AdoPrimaryRS*** con el modificador ***WithEvents*** . Esta palabra clave es opcional y especifica que la variable declarada es una variable de objeto utilizada para responder a eventos desencadenados por un objeto ActiveX. ***WithEvents*** solamente es válido en módulos de clase. Puede declarar tantas variables individuales como desee mediante ***WithEvents***, pero no puede crear matrices con ***WithEvents***. No puede utilizar ***New*** con ***WithEvents***.

Dentro del procedimiento ***Form_load()*** se declara una variable de tipo ***Connection*** con el nombre de *db*, la misma que mas adelante se inicializa una conexión, luego se observa en el código que la variable *db* ubica un cursor en el lado del cliente. A continuación, la variable *db* abre la conexión hacia un origen de datos (DSN) detallado en la cadena de conexión que tiene como argumento. El paso siguiente que se observa es como la variable recordset se inicializa y luego abre el conjunto de registros especificado con una sentencia SQL, la variable de conexión y; el tipo de recordset a abrir y el método de bloqueo de los registros. Hasta esta parte, el objeto recordset ya esta abierto y listo para ser manipulado.

La segunda manera de conexión a la base de datos es utilizando controles visibles de ADO, el mismo que se indica en la figura 6.26 (previamente se debe adicionar la

referencia Microsoft ADO Data Control 6.0 (SP4) (OLE DB) paravisualizar este contrtol en la barra de herramientas).

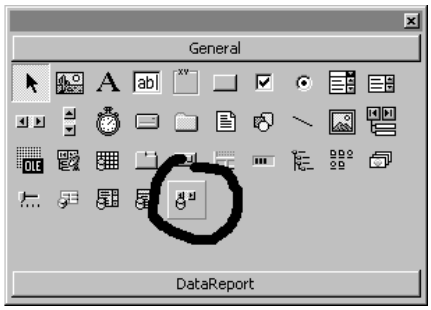


Figura 6.26

El control de acceso de datos del formulario de la figura 6.27 corresponde a un control ADO (ADOC).

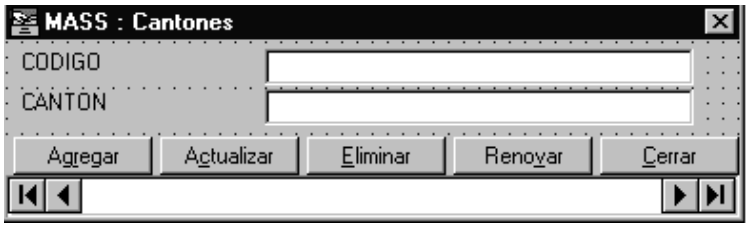


Figura 6.27

El Control de datos ADO es similar al control intrínseco **Data** y al **Control de datos remotos** (RDC). El **Control de datos ADO** permite crear rápidamente una conexión con una base de datos mediante **Objetos de datos ActiveX de Microsoft** (ADO).

Es posible crear en tiempo de diseño una conexión al establecer la propiedad **ConnectionString** con una cadena de conexión válida y, a continuación, la propiedad **RecordSource** con una instrucción apropiada para el administrador de base de datos. Puede establecer también la propiedad **ConnectionString** con el nombre de un archivo que defina una conexión; el archivo se genera mediante el cuadro de diálogo **Vínculo de datos** que aparece cuando hace clic en **ConnectionString** en la ventana Propiedades y, después, en **Generar** o en **Seleccionar**.

Conecte el **Control de datos ADO** a un control enlazado a datos como **DataGrid**, **DataCombo**, **DataList** o cuadros de texto; para esto, establezca la propiedad **DataSource** con el **Control de datos ADO**.

En tiempo de ejecución, establezca dinámicamente las propiedades **ConnectionString** y **RecordSource** para cambiar la base de datos. Opcionalmente, puede establecer la propiedad **Recordset** directamente con un conjunto de registros abierto previamente. Estos valores de configuración pueden ser ingresados en la ventana de propiedades de Visual Basic, tal como se muestra en la figura 6.28.

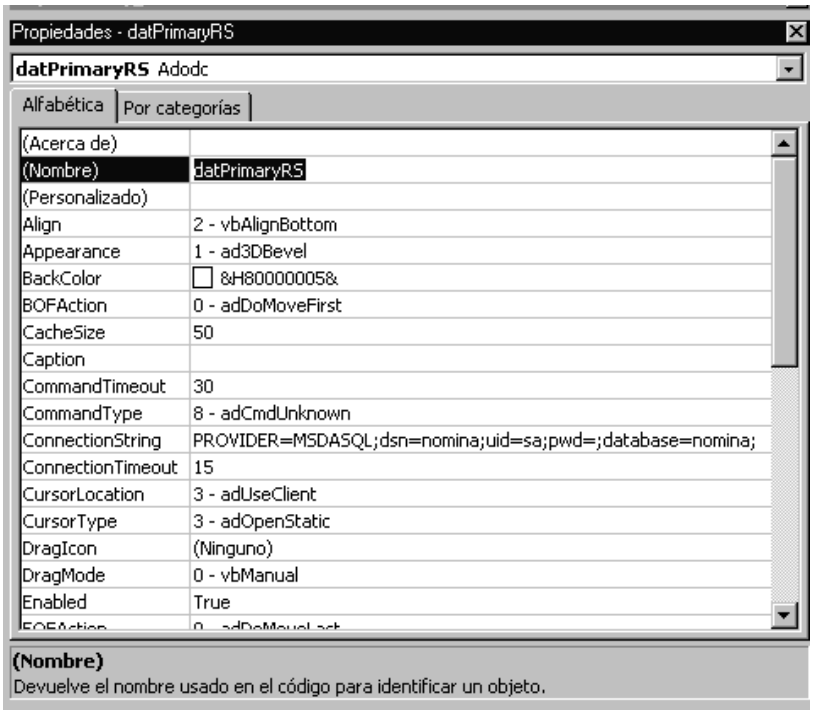


Figura 6.28

La tercera forma de conexión de datos utilizada en el prototipo es los objetos ***DataEnvironment*** que se utilizan en el proyecto básicamente para interactuar con los objetos ***DataReport***. Los controles ***DataEnvironment*** permiten agregar conexiones (connection), comandos (command) y proceccdimientos almacenados (stored procedures) para que puedan ser utilizados como una fuente de datos en cualquier aplicación. En primera instancia se deberá agregar el objeto ubicando para esto la opción ***proyecto*** y luego ***Agregar DataEnvironment***, de las forma tal como lo muestra la figura 6.29.

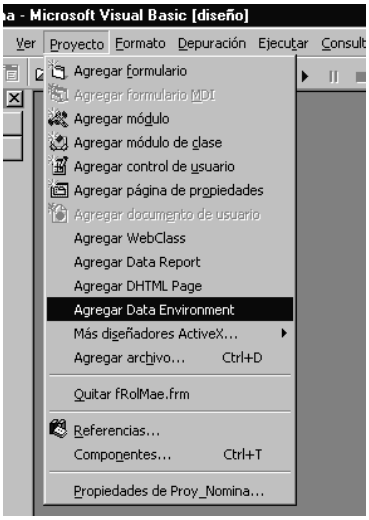
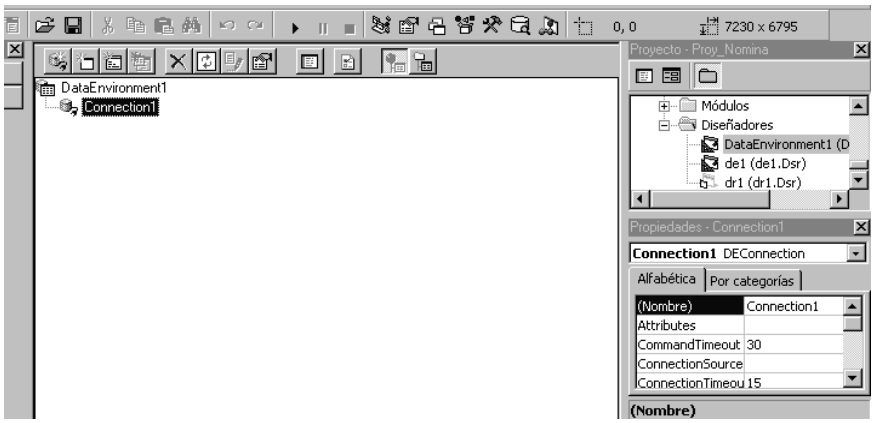


Figura 6.29

Ahora, debe aparecer una ventana como la mostrada en la figura 6.30, en esta ventana se procede a la configuración de la conexión.

Figura 6.30



Para configurar la conexión deberá hacer click con el botón derecho del mouse sobre el objeto ***Conexion1*** tal como se muestra en la figura 6.31.

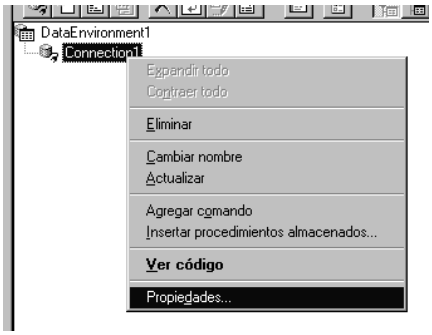


Figura 6.31

Para iniciar la configuración debe aparecer el cuadro de dialogo de la figura 6.32, e iniciar configurando el proveedor de la conexión, para el caso de ASE Sybase 11.5 se deberá escoger ***Microsoft OLE DB Provider for ODBC Drivers*** .

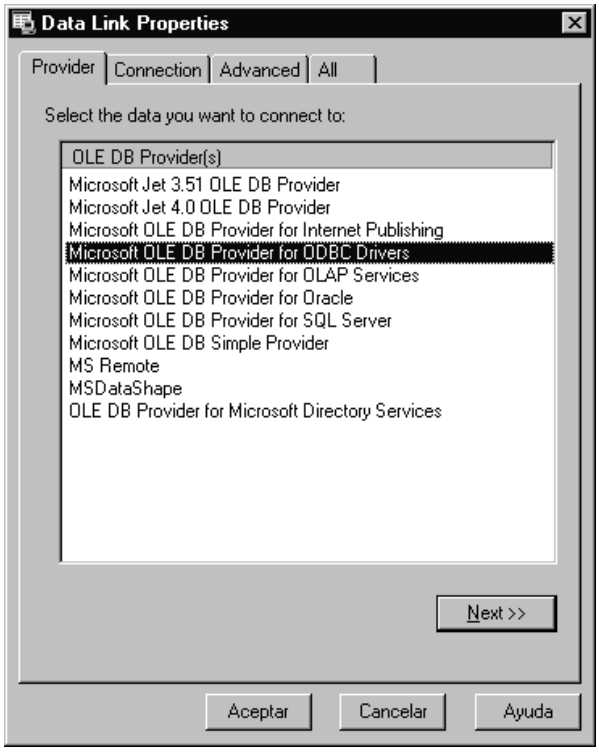


Figura 6.32

Luego de escoger el proveedor se procede a configurar la conexión propiamente dicha, aquí (se ve en la figura 6.33) se debe indicar el nombre del origen de datos (Data Source name) como un DSN previamente definido o como una cadena de conexión ODBC. Luego se deberá indicar el nombre de usuario y la contraseña validas para

ingresar a la base de datos, si existe un catálogo inicial se lo deberá mencionar en la parte inferior donde se solicita.

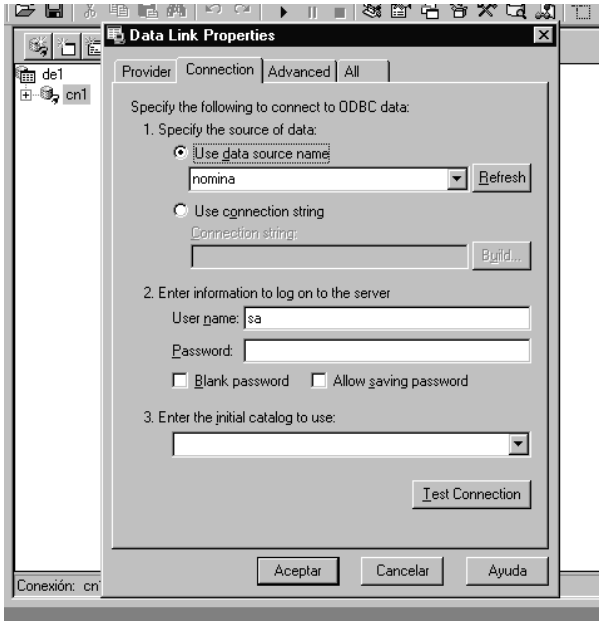


Figura 6.33

El siguiente paso consiste en probar si existe conexión al DSN mencionado, para esto se debe hacer click con el mouse en el botón <Test Connection> que se muestra en la figura 6.34, el resultado de esta prueba se mostrará en un cuadro de mensaje como el que se indica en la figura NN.

Figura 6.34

Con el paso anterior concluido ya se puede trabajar un una aplicación, pero existen todavía dos viñetas en la configuración de la conexión del ***DataEnvoironment*** . Si el proveedor OLEDB lo permite se pueden agregar otros datos de inicialización en la viñeta ***Advanced Tab*** , es decir, estos datos son específicos a un DBMS. En el caso de ***Sybase*** (figura 6.35) es permitido ingresar un ***Connect Timeout*** que indica el

tiempo en segundos que el proveedor OLEDB espera hasta completar la inicialización, si este tiempo expira retorna un mensaje de error y no se crea la conexión. También se puede señalar los privilegios sobre los datos que tendrá esta conexión en particular, solamente colocando una marca en los check box que se muestran.

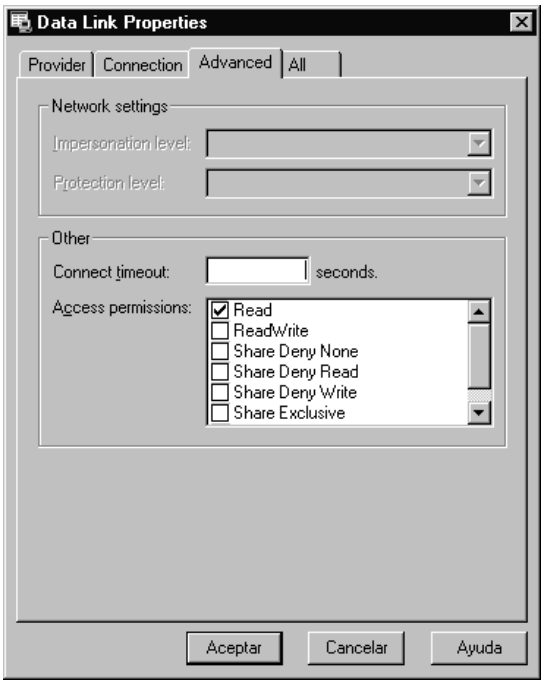


Figura 6.35

Finalmente, se puede visualizar y editar las propiedades de conexión en la viñeta ***All*** (tal como se muestra en la figura 6.36), aquí se pueden observar todos los parámetros de conexión.

Las formas de conexión mencionadas son las utilizadas estrictamente en el prototipo y el procedimiento de conexión y los parámetros establecidos obedecen al requerimiento del mismo.

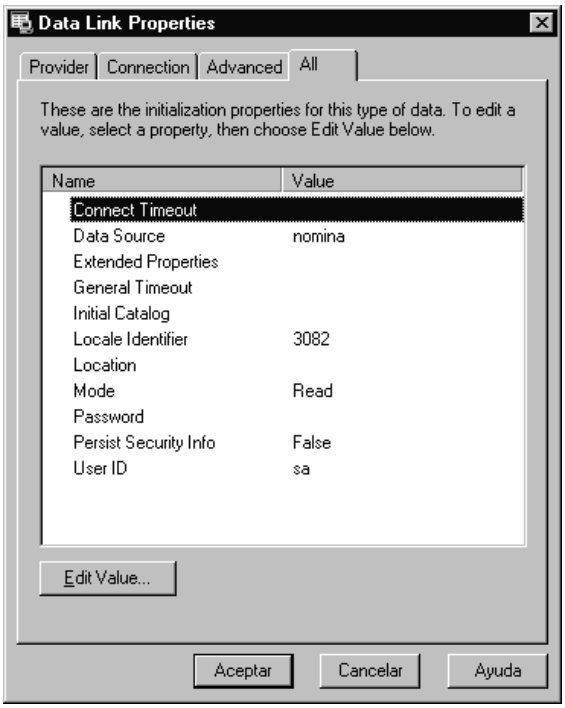


Figura 6.36

6.1.13.18. Documentos de Diseño

Los documentos de diseño se pueden encontrar en los siguientes archivos :

Diseño de Back-End (Base de datos) :

- Nomina.cdm
- Nomina.pdm
- Nomina.sql
- Insert.sql

Diseño de Front-End (prototipo)

- DataReport1.DCA
- DataReport1.Dsr
- DataReport1.dsx
- de1.DCA

- de1.Dsr
- de1.dsx
- dr1.DCA
- dr1.Dsr
- dr1.dsx
- fAccPer.frm
- fAccPer.frx
- fAcercaDe.frm
- fAcercaDe.frx
- fActUsuarios.frm
- fActUsuarios.frx
- fCanton.frm
- fCanton.frx
- fCiudad.frm
- fCiudad.frx
- fColegio.frm
- fColegio.frx
- fCuentas.frm
- fCuentas.frx
- fDatInfGen.frm
- fDatInfGen.frx
- fDatInfGen1.frm
- fDatInfGen1.frx
- fDepartamentos.frm
- fDepartamentos.frx
- fDestruncar.frm
- fDestruncar.frx
- fEscogePer.frm
- fEscogePer.frx
- fEscoger.frm
- fEscoger.frx
- fEscuelas.frm
- fEscuelas.frx

- fEstCivil.frm
- fEstCivil.frx
- fInicio.frm
- fInicio.frx
- fLogin.frm
- fLogin.frx
- fmdiPpal.frm
- fmdiPpal.frx
- fMesPago.frm
- fMesPago.frx
- fMesProces.frm
- fMesProces.frx
- Form1.frm
- fPais.frm
- fPais.frx
- fParroq.frm
- fParroq.frx
- fPeriodo.frm
- fPeriodo.frx
- fProfesiones.frm
- fProfesiones.frx
- fProvin.frm
- fProvin.frx
- fPuestos.frm
- fPuestos.frx
- frmMain.frm
- fRolDet.frm
- fRolDet.frx
- fRoles.frx
- fRolIndiv.frm
- fRolIndiv.frx
- fRolMae.frm
- fRolMae.frx

- fRubros.frm
- fRubros.frx
- fSexo.frm
- fSexo.frx
- fTipoAccPer.frm
- fTipoAccPer.frx
- fTipoEnte.frm
- fTipoEnte.frx
- fTipoPago.frm
- fTipoPago.frx
- fTitulos.frm
- fTitulos.frx
- fUniversidad.frm
- fUniversidad.frx
- fUsuarios.frm
- fUsuarios.frx
- LISTA.TXT
- local.ico
- massnom.ico
- mFunciones.bas
- MSSCCPRJ.SCC
- mVarGlob.bas
- proceso.ico
- Project2.vbw
- Proy_Nomina.vbp
- Proy_Nomina.vbw
- remoto.ico

6.2. Referencia operativa

En esta sección se vera la forma de operar el prototipo del aplicativo, teniendo en cuenta ciertas consideraciones desde el punto de vista de las herramientas utilizadas.

6.2.1. Concepción general del diseño

El diseño de la interfaz de usuario se ha concebido tomando en cuenta las normas básicas de una aplicación Cliente/Servidor, la interfaz propuesta tiene la siguiente estructura:

- *Conexión* .- Submódulo que permite conectarse a la base de datos autentificando un nombre de usuario y una contraseña que guardan una configuración especifica para ese usuario en particular. El prototipo dependiendo del valor de los parámetros puede negar el acceso al mismo o de manera contraria puede permitir iniciar una sesión con los permisos concedidos a ese usuario. Este modulo permite también guardar el histórico de los ingresos y egresos de los usuarios.
- *Catálogos* .- Este submódulo permite ingresar los valores de los catálogos o parámetros iniciales para arrancar una sesión con el aplicativo. Es decir, en esta parte se puede ingresar valores iniciales tales como: ciudades, parroquias, tipos de pago, departamentos, etc., valores que son necesarios para iniciar una sesión con el prototipo.
- *Mantenimiento* .- Permite mantener las tablas de roles de pago por periodo tanto de un empleado como de los valores generales para toda la nomina. Se pueden agregar, eliminar o modificar registros de las tablas de mayor movimiento del modelo.
- *Procesos* .- Aquí se pueden realizar los procesos masivos del prototipo, es decir, se puede procesar o generar los registros de toda la nomina en general bajo una condición especifica que se aplica a todos y cada uno de los empleados.
- *Reportes* .- Los reportes son listados imprimibles masivos de toda la nomina en general.
- *Consultas* .- Las consultas están concebidas como preguntas puntuales a la base de datos de acuerdo a la necesidad de los usuarios, puede imprimirse el resultado.

6.2.2. Administrador de Base de Datos

Para administrar la base de datos se puede utilizar directamente los wizards del Sybase Central en lo que se refiere a Creación, eliminación o actualización de bases de datos,

tablas, campos, etc. Además, se puede utilizar la herramienta ***Isqlw*** para mantener la base de datos tanto en estructura como en datos a través de la manipulación de scripts de ***TransacSQL***. La instalación de esta herramienta ya se indico en la sección *6.1.8. Instalación del Servidor y las herramientas de cliente* . Se puede administrar la base de datos utilizando la herramienta ***SQL Advantage*** propia de Sybase que aparece en las utilidades del Sybase Central. Esta utilidad es similar a ***Isqlw***, pero se recomienda utilizar esta última.

6.2.3. Plataforma de Servidor, Cliente y Red

La estructura global del prototipo para administración de Nómina de Personal ha sido probado efectivamente sobre una plataforma con las siguientes características:

Motor del servidor de datos	: *Sybase Adaptive Server 11.5 for Windows NT*
Software de servidor de datos	: *librerías de servidor ASE Sybase 11.5*
Nombre del servidor de datos	: *Development*
Sistema Operativo de Servidor	: *Microsoft Windows NT Server 4.0*
Service Pack	: *SP 6*
Option Pack	: *OP 2*
Sistema Operativo de Clientes	: *Microsoft Windows 95/98/Me/NT*
Software de cliente de datos	: *librerías de cliente ASE Sybase 11.5*
Data Source Name	: *nomina*
Nombre de Base de Datos	: *nomina*
Software Development Kit	: *Microsoft Visual Studio 6.0 / MS Visual Basic 6.0*

6.2.4. Usuarios y niveles de seguridad

El conjunto de opciones para el árbol de menús consta de 255 opciones, es decir, se pueden obtener el resultado de 255 combinaciones diferentes. Este numero es bastante grande y solo depende del requerimiento, la implementación de este y la creación del rol por parte del administrador de la Base de Datos, utilizando la siguiente instrucción SQL :

```
insert into TROLES values ("1","ADMINISTRADOR",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1")
```

Donde cada "1" significa la habilitación de una opción en el árbol de menús de acuerdo a la estructura de la tabla ***tRoles*** y de la nominación del ítem de menú en la implementación del aplicativo. Una inhabilitación de una opción corresponde a un valor de "0". Entonces los roles y sus nombres son proporcionales al numero de combinaciones en la tabla ***tRoles***. Existe una tabla de Usuarios ***tUsuario***, la misma que guarda los datos básicos de los posibles usuarios del sistema y a partir de esta se pueden agregar a la tabla ***tLogin*** donde se guardan los datos de los usuarios que tienen una cuenta de ingreso al sistema con su respectivo rol, y un campo de estado que permite en cualquier momento inhabilitar esa cuenta en particular. De igual manera, el numero de usuarios y cuentas de usuario depende del numero de ingresos a sus respectivas tablas, es decir no existe un límite. El grafico 6.37 indica donde se puede agregar nuevos usuario y crear sus cuentas, véase <Usuarios..> y <Cuentas..>, además la opción <Cambiar Estado de Usuario...> permite habilitar o inhabilitar una cuanta de usuario (refiérase al grafico 6.38) y la opción <Destruncar Usuario...> permite rectificar un fin de sesión erróneo y un registro incompleto en la tabla ***tLog*** (refiérase al grafico 6.39). En el grafico se puede ver la manera de escoger una cuenta de usuario a destruncar, esta opción esta habilitada solo para quienes tienen el rol de ***Adimistrador***.

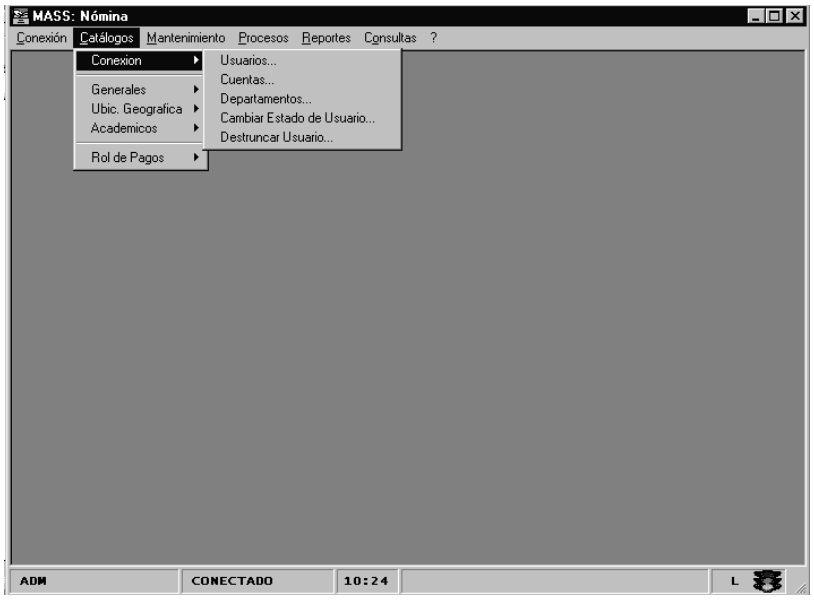


Figura 6.37

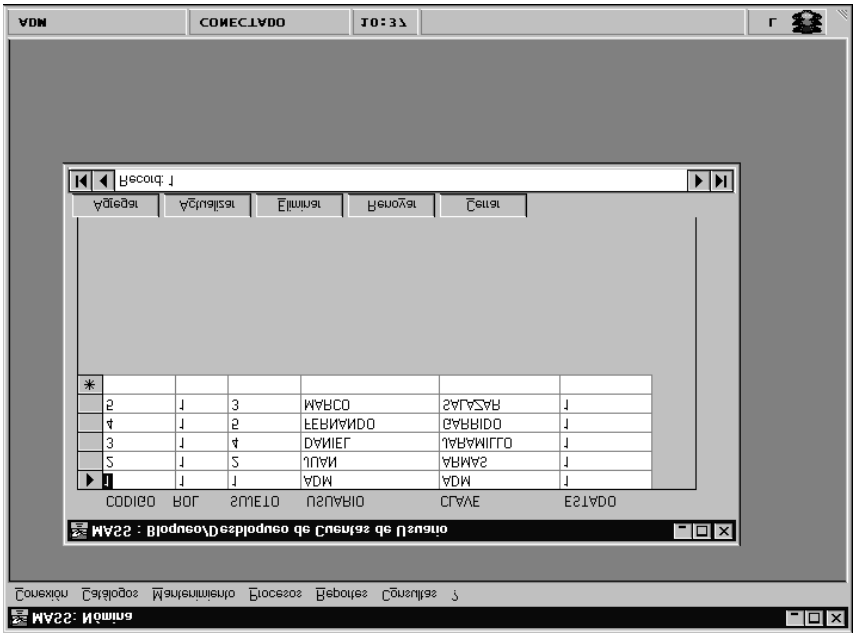
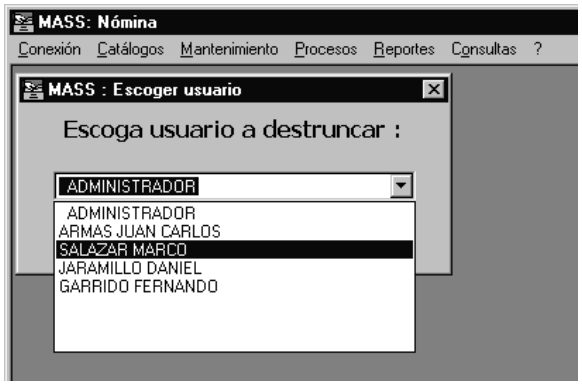


Figura 6.38



Fgura 6.39

luego de escoger la cuenta de usuario a destruncar, si esta cuenta tiene una salida errónea, aparecerá un formulario con un campo **FIN DE SESION : 1/1/1900** , la misma que debe cambiarse por la fecha actual del cambio. Si no existiera una sesión mal finalizada, el formulario aparece vació. Refiérase a la figura 6.40.

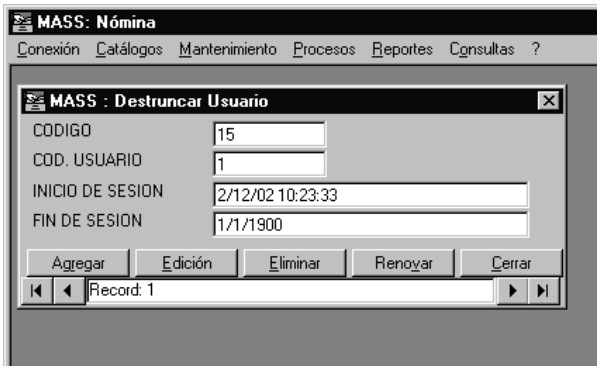


Figura 6.40

6.2.5. Logon y logout

Todos los usuarios deben iniciar una sesión con el sistema, esto es, deben realizar un ***logon*** en el frontal para ingresar a las opciones habilitadas de acuerdo a su rol (véase la figura 6.41), de esta manera se asegura que los accesos sean autorizados, la correcta apertura y cierre de una conexión con la base de datos y entre otras ventajas, el registro del inicio de sesión que permita en el futuro aplicar consultas estadísticas o procesos auditables de los accesos al sistema.

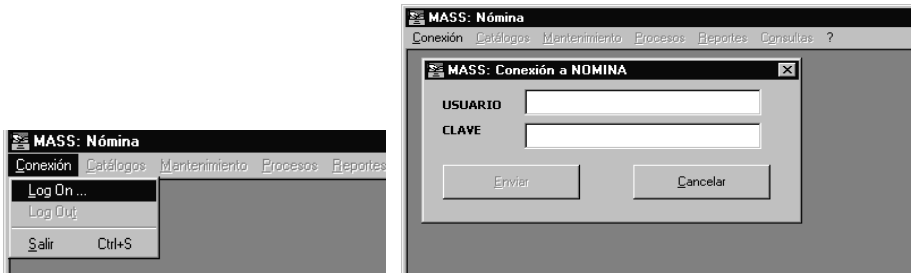


Figura 6.41

Para abandonar el sistema, el usuario debe realizar un ***logout***, el mismo que cierra de manera correcta la conexión a la base de datos y registra el final de la sesión en la tabla ***tLog*** , se deberá esperar el cuadro de dialogo que se muestra en la figura 6.42, datos que se utilizan para realizar procesos futuros. En caso de que algún usuario no realice el ***logout*** y cierre la aplicación por otro medio no se registra el fin de la sesión y la próxima sesión que intente iniciar será rechazada, teniendo entonces el usuario que solicitar una autorización y cambio de estado al Administrador de la Base de Datos.

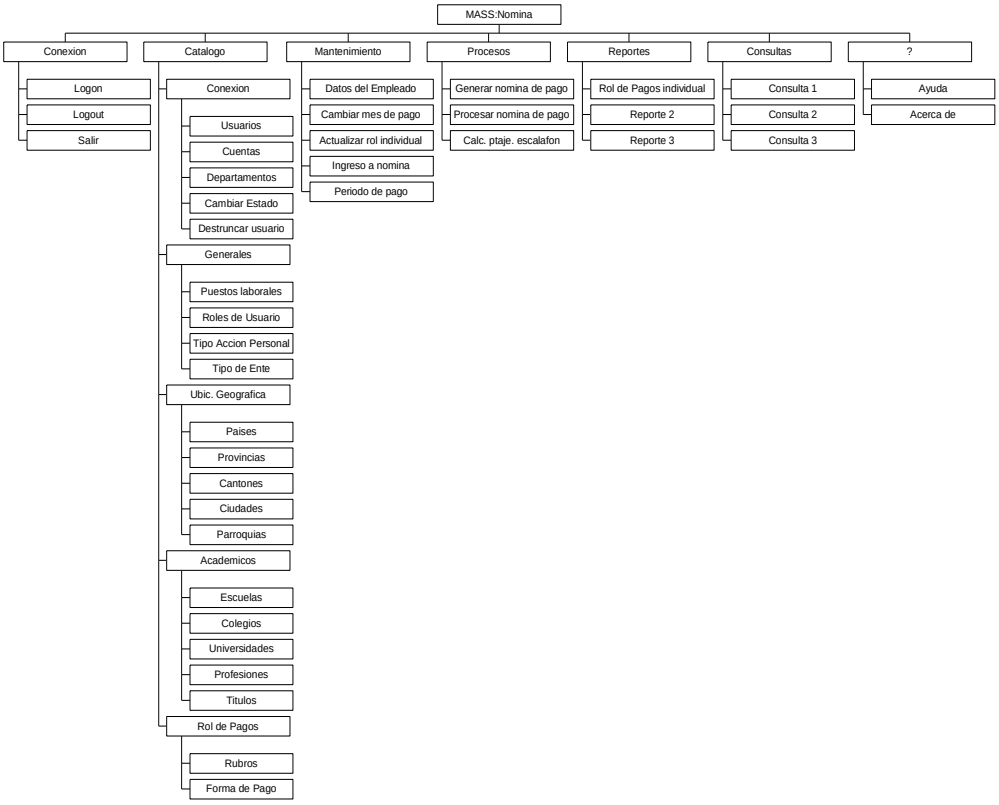
Figura 6.42

6.2.6. Estadísticas y Auditoría

Existe una tabla con el nombre de ***tLog***, la misma que contiene un historial de ingresos, salidas, usuario, fecha y hora de la sesión que pueden ser utilizados para realizar consultas estadísticas y también para aplicar procesos auditables del acceso al sistema, tomando en cuenta que cada usuario maneja un rol diferente que indica cuales son sus privilegios. Se puede revisar la estructura de la tabla ***tLog*** para conocer sus atributos y las relaciones que permitan realizar las operaciones indicadas.

6.2.7. Árbol de menús

El esquema de lo menús del aplicativo tiene la siguiente estructura:



6.2.8. Operación del prototipo

La operación del prototipo tiene un patrón estándar, es decir, los formularios funcionan de manera similar con barras de botones de comando que realizan básicamente el mismo trabajo en todas las instancias.

6.2.8.1. Inicio de sesión

Cualquier usuario deberá iniciar una sesión conectándose a la base de datos a través de la opción **[Conexión]** en el ítem **[Login]** , donde se le pedirá un nombre de usuario y una clave, si el usuario esta bloqueado por el Administrador, truncado por una operación inapropiada o no tiene ciertos privilegios no podrá iniciar una sesión, caso contrario inicia una sesión con los ítems y los privilegios que su rol le proporcionen, además la ventana de la aplicación indica el usuario, la hora y el estado de conectado (ver Figura 6.43). Esta operación de login y logout ya esta documentada en la sección ***6.2.5. Logon y Logout*** y la sección ***6.2.4. Usuarios y Niveles de***

Seguridad. Para finalizar una sesión de forma normal deberá elegir la opción **[Conexión]** en el ítem **[Logout],** misma que actualiza el registro de login en la tabla ***tLog*** para futuras estadísticas y auditorias.

Figura 6.43

Existe un caso especial donde ningún usuario puede iniciar una sesión, este es cuando no existe un periodo de trabajo establecido previamente, esto se puede dar en el caso de que el Administrador haya modificado el campo ***PER_ESTADO*** de la tabla ***TPERIODO***. Si esto ocurre aparecerá un mensaje de error al iniciar la aplicación solicitando que se comunique con el Administrador, el Administrador a su vez debe actualizar la tabla ***TPERIODO*** ejecutando el siguiente script SQL :

```
use nomina
go
update TPERIODO
set PER_ESTADO = "A"
where PER_COD = n
```

donde **n** es el código (clave principal) del registro que se va a actualizar en la tabla TPERIODO. Hecho esto ya se podrá iniciar una sesión.

6.2.8.2. Operación básica

Antes de iniciar a detallar la operación básica es necesario indicar que todos los formularios responden a un diseño Cliente/Servidor que permite mantener un recordset en el Cliente de manera exclusiva, bloqueando los registros en el servidor hasta que sean liberados, además, todo el procesamiento se realiza en el Cliente debido a las características del hardware de este.

Existen dos barras de botones de comando que se pueden encontrar en los formularios del prototipo, las mismas que se pueden ver en las figuras 6.44 y 6.45, para referirse a ellas se las nombrara como Barra 1 (Figura 6.44) y Barra 2 (Figura 6.45).

La Barra 1 obedece a un diseño del formulario basado en controles ADO (ActiveX Data Control) y cada uno de los botones de la barra funcionan de la siguiente manera:

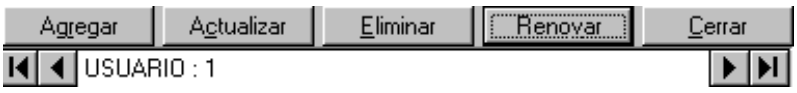


Figura 6.44

- ***Agregar*** .- Permite agregar un nuevo registro a la tabla correspondiente al formulario, si se desea ingresar un nuevo ítem primeramente se deberá crear el registro en blanco para luego ingresar los datos correspondientes.
- ***Actualizar*** .- Cualquier cambio en los datos del formulario se deben actualizar en la tabla presionando este botón.
- ***Eliminar*** .- Para eliminar un registro se debe presionar este botón, pero previo al posicionamiento efectivo en dicho registro, para esto se debe utilizar los botones de desplazamiento que se encuentran bajo la barra de botones.
- ***Renovar*** .- Corresponde a una operación de ***Refresh*** para realizar transacciones en ambientes multiusuario, es decir, este botón permite refrescar los datos del formulario desde la base de datos directamente.
- ***Cerrar*** .- Permite cerrar el formulario actual, cuando haya mas de un formulario este puede cerrarse en cualquier orden (no son ***Modales***) debido a

que el diseño los ha contemplado con la propiedad ***MDIChild***, es decir, son parte de un formulario padre ***FMDI*** .

La Barra 2 obedece a un diseño del formulario basado en objetos ADO (código de ADO), debido a que facilita la manipulación del código interno de administración de los datos. Cada uno de los botones de la barra funcionan de la siguiente manera:

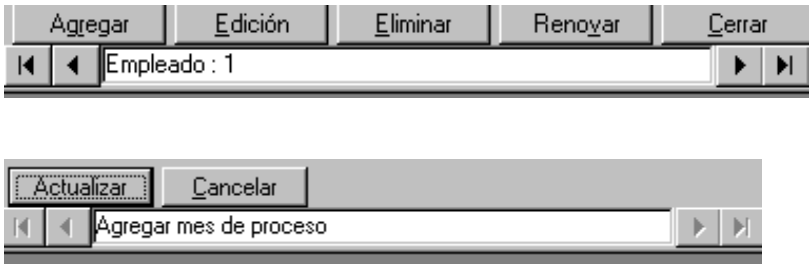


Figura 6.45

- ***Agregar*** .- Permite agregar un nuevo registro a la tabla correspondiente al formulario, si se desea ingresar un nuevo ítem primeramente se deberá crear el registro en blanco para luego ingresar los datos correspondientes. Este botón se convierte en el botón **[Actualizar]** una vez que ha sido presionado.
- ***Actualizar*** .- Sirve para guardar los datos ingresados o para actualizar cualquier cambio en los datos del formulario.
- ***Edición*** .- Permite pasar al modo edición de datos en el formulario correspondiente, primero se debe editar los datos para luego actualizarlos. Una vez presionado este botón se convierte en el botón **[Cancelar].**
- ***Cancelar*** .- Cancela las acciones de agregar o editar y regresa al estado anterior.
- ***Eliminar*** .- Para eliminar un registro se debe presionar este botón, pero previo al posicionamiento efectivo en dicho registro, para esto se debe utilizar los botones de desplazamiento que se encuentran bajo la barra de botones.
- ***Renovar*** .- Corresponde a una operación de ***Requery*** para realizar transacciones en ambientes multiusuario, es decir, este botón permite refrescar los datos del formulario desde la base de datos directamente.
- ***Cerrar*** .- Permite cerrar el formulario actual, cuando haya mas de un formulario este puede cerrarse en cualquier orden (no son ***Modales***) debido a

que el diseño los ha contemplado con la propiedad ***MDIChild***, es decir, son parte de un formulario padre ***FMDI*** .

El funcionamiento de estas dos barras de botones es el mismo para los formularios que las contengan, su operación se basa en el concepto Cliente/Servidor en donde los eventos deben seguir un orden lógico debido al bloqueo de los registros.

6.2.8.3. Tour a través el prototipo

Figura 6.46

Para iniciar una sesión con el prototipo se debe hacer click sobre el icono que se muestra en la Figura 6.46.

Luego aparecerá la ventana de splash del prototipo (se muestra en la Figura 6.47).

Figura 6.47

A continuación, en la ventana principal del prototipo se debe conectarse a la base de datos en **[Login]** tal como se muestra en la figura 6.48.

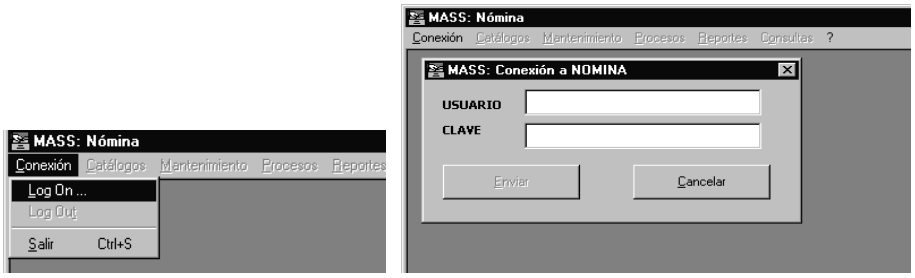


Figura 6.48

Debe seguir el proceso indicado en la sección **6.2.8.1 Inicio de sesión**, luego de haber ingresado a la aplicación se puede empezar a revisar cada uno de los ítems del menú principal.

a. Catálogos

- ***Conexión*** .- Opciones que permiten administrar los parámetros de usuarios nuevos y antiguos. Los valores del código en todos los formularios son automáticos, es decir, el usuario esta exento de ingresarlos.
  - *Usuarios* .- Ingreso y mantenimiento de usuarios.
  - *Cuentas* .- Creación y mantenimiento de cuentas de usuario. Una cuenta activa debería tener el parámetro ESTADO = 1, caso contrario con el valor 0.
  - *Departamentos* .- Ingreso y mantenimiento de departamentos de la institución
  - *Cambiar estado de usuario* .- Permite visualizar un listado general de los usuarios y sus cuentas para cambiar el estado de los requeridos, la columna ESTADO puede cambiar a los valores 1= ACTIVO o 0 = INACTIVO. Tal como se muestra en la figura 6.49.

Figura 6.49

- *Destruncar usuario* .- Permite destruncar a un usuario cuando este no ha realizado un Logout a la base de datos luego de iniciar una sesión, en primer lugar se debe escoger el usuario como se muestra en la figura 6.50. Luego se deberá presionar el botón continuar y aparece la ventana que se ve en la figura 6.51. En el cuadro de texto FIN DE SESION se deberá cambiar la fecha que aparece originalmente a la fecha actual y luego actualizar el registro.

Figura 6.50

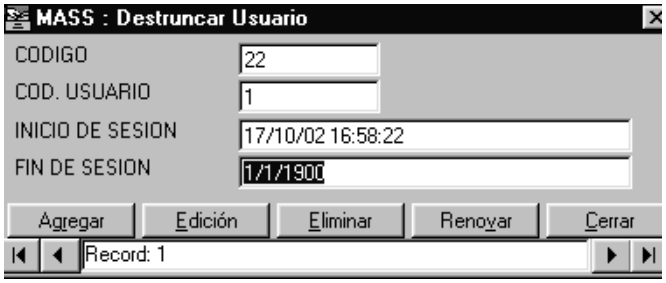


Figura 6.51

- ***Generales*** .- Son valores de parámetros existentes y nuevos para catalogar otras entidades del modelo.
  - *Puestos laborales* .- Permite dar mantenimiento a las ubicaciones laborales existentes y antiguas que soporte la institución.
  - *Roles de usuario* .- Permite mantener los roles de usuario nuevos o antiguos, cada rol puede manejar una combinación de 254 opciones que corresponden a los ítems del árbol de menús.
  - *Tipos de acción de personal* .- Permite dar mantenimiento a las acciones de personal que se establezcan en la institución.
  - *Tipo de Ente* .- Permite mantener el catalogo de los tipos de entes que la entidad clasifique (ej. Administrativo, Trabajador, etc).

- ***Ubicación Geográfica*** .- Son valores de parámetros existentes y nuevos. Los códigos para estos registros son de tipo alfanumérico correspondiente a la normalización mundial de ubicaciones geográficas. En el Ecuador estos valores pueden ser provistos por el INEN (Instituto ecuatoriano de Normalización).
  - *Países*
  - *Provincias*
  - *Cantones*
  - *Ciudades*
  - *Parroquias*

- ***Académicos*** .- Son formularios para dar mantenimiento a valores de parámetros académicos existentes y nuevos que son necesarios para crear el currículo de los empleados.
  - *Escuelas*
  - *Colegios*
  - *Universidades*
  - *Profesiones*
  - *Títulos*

- ***Rol de pagos*** .- Son formularios para dar mantenimiento a los parámetros de la forma de pago y la interfaz contable de los rubros existentes y nuevos.

- *Rubros* .- Permite ingresar los rubros que la institución haya establecido para el rol de pagos. En la figura 6.52 se puede observar que cada rubro tiene una CTA CONTABLE correspondientes a su par en el plan de cuentas de la institución, además una marca de D = DEBE o H = HABER y una marca de I = INGRESO o E = EGRESO para identificar el rubro en el balance.

Figura 6.52

- *Forma de pago* .- Permite mantener los registros de periodicidad de pagos a los empleados.

b. Mantenimiento

- ***Datos del empleado*** .- Permite ingresar datos de identificación y ubicación general del empleado, algunos valores sirven para cálculos posteriores. La mayoría de datos se obtienen a través de cuadros combinados (como se ve en la figura 6.53) que despliegan los catálogos antes ingresados.

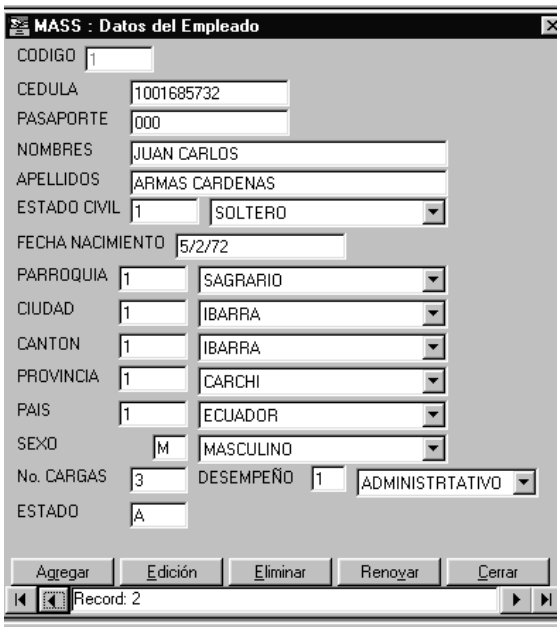


Figura 6.53

- ***Cambiar mes de pago*** .- Esta opción permite cambiar el mes de pago activo de una lista de todos los meses ingresados para motivos de recalculo, arreglo de meses anteriores o simplemente para procesar un nuevo periodo, la ventana se muestra en la figura 6.54.

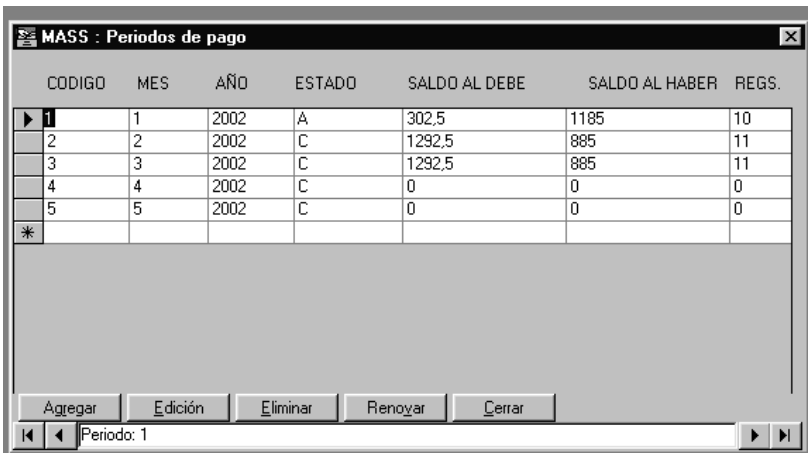


Figura 6.54

Una vez cambiado el mes de proceso en modo edición se debe proceder a actualizar los cambios, al momento de hacerlo debe aparecer una ventana como la de la figura 6.55 que verifica la acción a tomar.

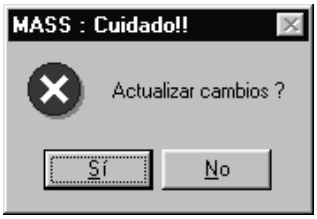


Figura 6.55

Si por alguna razón cambio a estado ACTUAL mas de un periodo mensual el aplicativo se asegura y presenta un mensaje de error como el de la figura 6.56.

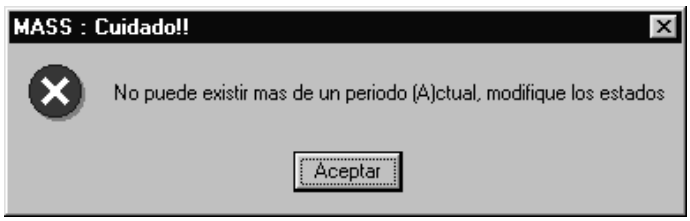


Figura 6.56

El caso contrario seria si no existe en la tabla ningún periodo mensual en estado ACTUAL, en este caso también aparece un mensaje de error igual al de la figura 6.57.

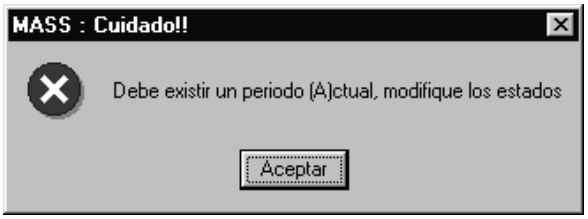


Figura 6.57

- ***Actualizar rol individual*** .- Este formulario permite realizar cambios sobre el rol individual de cada empleado en el periodo mensual ACTUAL, es decir, si a un empleado se le debe retirar o aumentar un rubro en la nomina se debe primero ubicarlo y luego realizar el cambio correspondiente en un formulario como el que se ve en la figura 6.58, cabe mencionar que la actualización del rol es párale mes ACTUAL, los periodos anteriores permanecen con los rubros que mantenian anteriormente.

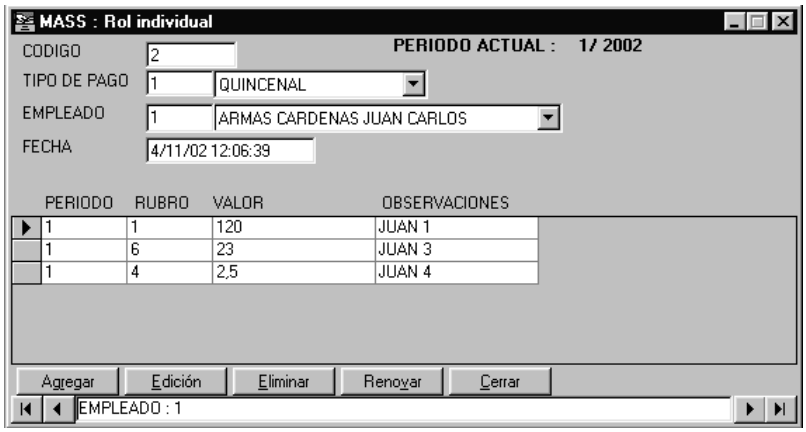


Figura 6.58

- ***Ingreso a nomina*** .- Este formulario se debe utilizar para crear la nomina inicial de la institución o para ingresar nuevos empleados a la misma, nótese en la figura 6.59 que en primer lugar se escoge un empleado de los ya ingresados (o en su defecto primero se debería ingresar el empleado), luego se debería guardar los datos del nuevo ingreso y finalmente presionar el botón de comando [INGRESAR RUBROS].

Figura 6.59

Para ingresar los rubros del ingreso aparece un formulario como el de la figura 6.60, en donde se escoge el rubro, se ingresa el valor y se escribe una observación adicional si existiere.

Figura 6.60

- ***Periodos de pago*** .- Este formulario permite ingresar nuevos periodos mensuales de pago, para lo cual se debe ingresar solamente el mes y el año, el resto de parámetros de este formulario están previamente ingresados y solamente cambiaran el momento que este periodo sea ACTUAL y además sea procesado. Puede verse el formulario en la figura 6.61.

Figura 6.61

c. Procesos

- ***Generar nomina de pago*** .- El formulario permite escoger un periodo de pago posterior al ACTUAL (como se ve en la figura 6.62) para generar los registros de la nomina en la tabla TROLDET , el proceso interno que se realiza consiste en copiar el conjunto de registros del mes ACTUAL para el mes que se esta procesando, adicionalmente, si algún empleado necesita actualizar su rol

individual se puede realizar los cambios en el formulario **[ACTUALIZAR ROL INDIVIDUAL]**.

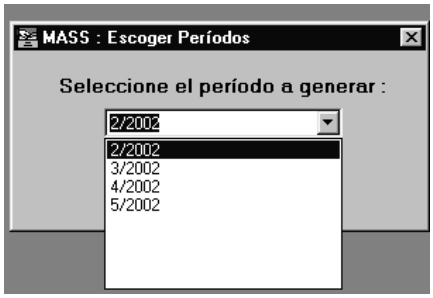


Figura 6.62

Luego de escoger el periodo a generar aparece una ventana de verificación como la de la figura 6.63 para reafirmar o cancelar la acción.

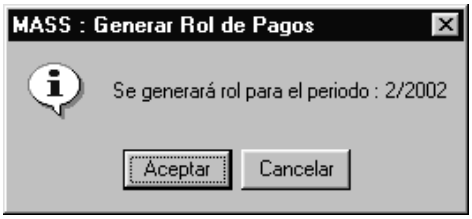


Figura 6.63

Si durante la generación no ocurrió ningún inconveniente, debe aparecer una ventana de mensaje como la de la figura 6.64.

Figura 6.64

Cabe indicar que para generar un mes de pago no es necesario que dicho mes este establecido como ACTUAL, este estado es necesario para realizar su procesamiento.

- ***Procesar nomina de pago*** .- La opción de procesamiento (ver Figura 6.65) permite realizar los cálculos de toda la nomina, tomando cada rol individual de uno en uno de acuerdo a las modificaciones realizadas individualmente si acaso existieron.

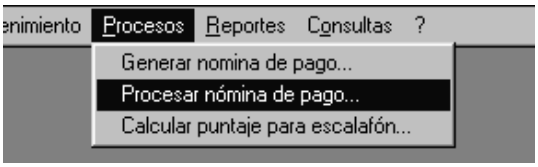


Figura 6.65

En la figura 6.66 se muestra el resultado del proceso de la nomina de pago, aquí se indica: el mes de proceso, el estado del mes, el saldo al debe, el saldo al haber, el total de valores como ingresos, el total de valores como egresos y el numero de registros procesados.

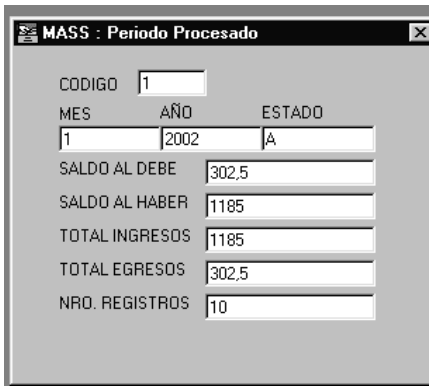


Figura 6.66

- ***Calcular puntaje para escalafón*** .- Permite calcular el puntaje individual para el escalafón interno de la institución.

d. Reportes

- ***Rol de pagos individual*** .- Permite visualizar la nomina general como boletines individuales, tal como muestra la figura 6.67.

Es posible crear otros reportes de acuerdo a la necesidad del usuario del prototipo.

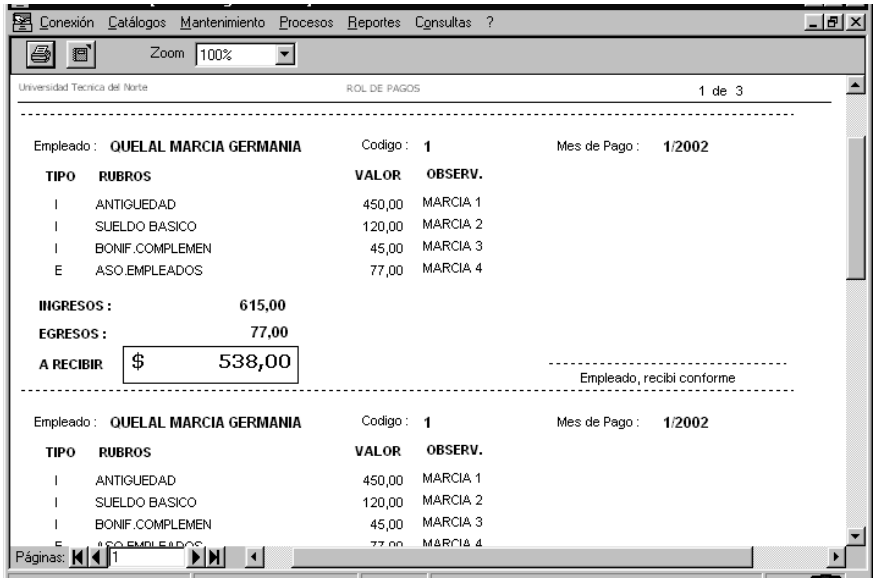


Figura 6.67

e. Consultas

En esta sección se pueden realizar consultas especificas cerrando el resultado a registros particulares.

6.2.8.4. Estado de los procesos

El estado del proceso de las opciones del menú se puede observar en la esquina inferior derecha de la ventana principal de la aplicación, en este sector se muestra un icono de un semáforo, en donde el color ROJO representa que el control lo tiene el SERVIDOR DE DATOS (ver figura 6.68),

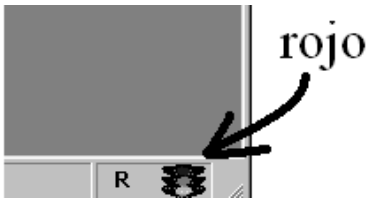


Figura 6.68

el color AMARILLO significa que el resultado esta en TRANSITO (ver figura 6.69), y,

Figura 6.69

el color verde significa que los resultados están en control del Cliente(ver figura 6.70).

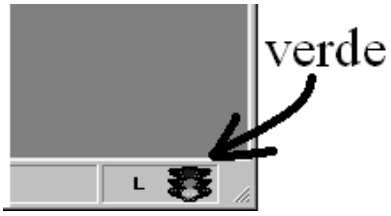


Figura 6.70