

## **CAPITULO II**



### **INTEGRACIÓN DE WINDOWS CON IBM AS/400**

- 2.1 Comparación de Características**
- 2.2 Integración al nivel de Conectividad**
- 2.3 Integración al nivel de Sistema Operativo**
- 2.4 Integración al nivel de BDMS**
- 2.5 Integración de AS/400 con Herramientas de Desarrollo Windows**

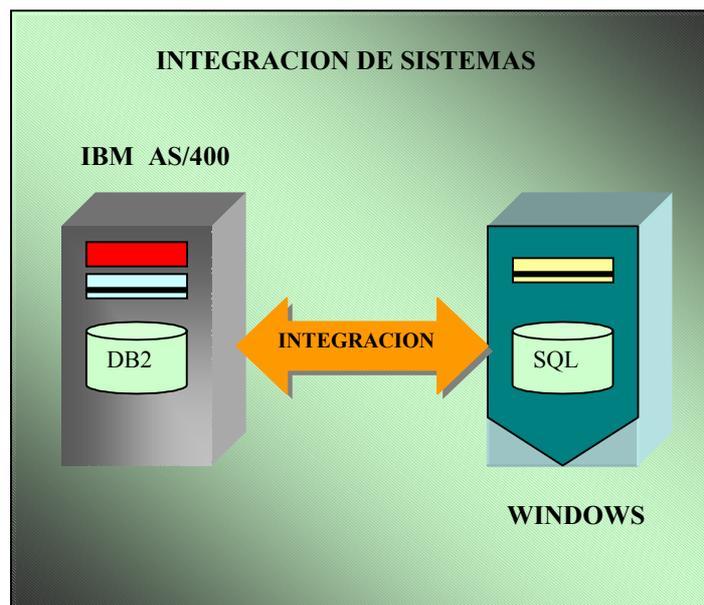
Una vez realizado un análisis completo del Sistema AS/400 de IBM en el capítulo anterior, se efectuó una comparación con la plataforma Windows de Microsoft sobre Computadoras Personales. Dicha comparación ha permitido verificar que las dos plataformas llegan a cumplir los parámetros necesarios para la integración de sus sistemas.

En la presente tesis no se encuentra información exhaustiva de Windows ya que es una plataforma muy conocida y su información es fácil de encontrar.

Las posibilidades de integración de AS/400 y Windows sobre Computadoras Personales se detallan a continuación en este capítulo.

Cabe manifestar que las dos casas comerciales de estas plataformas han tenido una comunicación constante en el diseño de software que las ha llevado a obtener beneficios de la integración de sus respectivos sistemas.

## 2.1 COMPARACIÓN DE CARACTERÍSTICAS



**Figura 2.1 Integración De IBM AS/400 Con Windows.**

Las dos plataformas ofrecen un diseño enteramente abierto a la comunicación con otros sistemas y con una clara intervención en las topologías más populares de red.

El sistema operativo de AS/400 es llamado OS/400, mientras que el sistema operativo de PC es Windows. La investigación del tema se realizó sobre la versión del sistema

operativo OS/400 V4R2(versión 4, release 2) en el modelo AS/400 150 y las versiones de Windows son 95, Windows 2000 sobre Computadores Personales.

A continuación se presenta un análisis comparativo de las principales características que tienen en común los dos sistemas operativos de estudio.

CARACTERÍSTICAS	SISTEMAS OPERATIVOS		
	OS/400	WINDOWS [LIB003]	OBSERVACIONES
Independencia de hardware	SI	SI	No fueron diseñados para un solo tipo de procesador.
Compatibilidad con multiprocesadores	SI	SI	Pueden tener más de un procesador
Multitarea y Multihebra	SI	SI	Ejecuta más de una tarea simultáneamente
Sistema operativo de red	SI	SI	
Seguridad	SI	SI	La seguridad del AS/400 es mucho mejor y muy renombrada.
Interfaz gráfica de usuario	NO	SI	AS/400 se vale de Windows para ofrecer interfaces gráficas en su gestión.
Procesadores	Power PC en diferentes modelos	Intel 80486, Pentium, DEC Alpha RISC, Power PC.	
Sistema de archivos jerárquico	SI	SI	En Windows FAT y NTFS En AS/400 Sistema de Archivos Integrado.
Protocolos	IPX/SPX, TCP/IP, SNA, APPC, APPN, HPR, UDP, OSI.	IPX/SPX, TCP/IP, NETBEUI, APPLE TALK, SNA DLC.	
Trabaja mediante Subsistemas	SI	SI	

**Tabla 2.1 Características De Windows Y AS/400.**

La tabla anterior muestra varias de las características que tienen en común los dos sistemas, con esto no se quiere decir que sean iguales. Es más, se puede afirmar que en su tecnología ni siquiera se parecen, cada una gestiona sus plataformas de manera distinta. Pero existen las condiciones necesarias para una comunicación amplia entre los dos sistemas.

Las dos plataformas basan sus tecnologías de comunicación en los estándares de protocolos de red principalmente en el estándar de transmisión de datos TCP/IP, además de sus propios protocolos.

Podemos analizar mejor la integración de AS/400 con Windows dividiéndola en tres niveles: nivel de conectividad, nivel de sistema operativo y nivel de aplicaciones.

## 2.2 INTEGRACIÓN DE NIVEL DE CONECTIVIDAD

Se ha denominado nivel de conectividad a la fase de comunicación de las dos plataformas en lo que se refiere a la conexión física y la capa física de protocolos de transmisión de información, ya sea en una conexión directa o dentro de un entorno de red.

Los sistemas AS/400 de IBM, perfectamente pueden intervenir en diferentes tipos de topologías de red de sistemas. Entre los diferentes tipos están los siguientes:

- Red en Anillo o Ethernet sobre TCP/IP.
- Asíncrona(SLIP) sobre TCP/IP.
- PPP(protocolo punto a punto) sobre TCP/IP.
- Red en Anillo sobre SNA(Arquitectura de red de Sistemas).
- Ethernet sobre SNA.
- Conexión Twinaxial sobre SNA.
- IPX(Intercambio de paquetes interredes), etc. **[LIB004]**

Con la información anterior se puede determinar que un sistema AS/400 puede estar conformando una red con más sistemas AS/400, esto lo hará poniendo énfasis en la comunicación con su propio protocolo de comunicaciones SNA. Si actúa dentro de una red con otros equipos o plataformas diferentes, como Windows la herramienta de transmisión de datos que se pondrá de manifiesto es el protocolo de redes TCP/IP.

El sistema AS/400 puede intervenir en una red LAN o WAN ya sea como servidor de archivos, servidor de base de datos, servidor de correo electrónico, servidor de aplicaciones, servidor de Data Warehouse, etc.

El sistema Windows también puede hacer de servidor de red aunque con menos posibilidades de gestión que el AS/400, esto si la versión es Windows Server.

Esta disposición del sistema AS/400 permite la comunicación con un PC con Windows mediante la red o conectado directamente mediante un adaptador como una estación de trabajo inteligente del AS/400, o como consola del sistema, esto gracias al software que aportan las dos plataformas para conseguirlo.

Los modos de conexión física de un AS/400 y un PC pueden ser los siguientes:

- Un PC Puede ser la consola de operación(Estación de trabajo ASCII) del sistema AS/400, eso implica que se puede conectar directamente a un Sistema AS/400 mediante un sistema de cableado IBM o cable Twinaxial.
- Otra manera de conexión es mediante un controlador para estación de trabajo, igualmente en modo de una Estación de trabajo ASCII.
- Otra manera de comunicación es utilizando una red de área local(LAN). El sistema AS/400 se conecta a la red utilizando dos tipos de adaptadores: adaptador Ethernet y adaptador Token-Ring. También mediante una red WAN en una conexión remota. **[DTP003]**.

TCP/IP debe ser instalado y configurado, independientemente en los dos sistemas para poder tener el intercambio de información. Los procedimientos en cada sistema son distintos. IBM, ofrece toda la información de estos procedimientos mediante manuales completos de la instalación y puesta a punto de TCP/IP para lograr la integración.

SNA gestiona las comunicaciones por la parte del OS/400, utilizando sus propios protocolos y seguridades. SNA es el protocolo más importante, propio del sistema AS/400, que sirve para la operación y configuración de las comunicaciones entre sistemas IBM y no IBM.

### **2.2.1 TCP/IP en el AS/400**

TCP/IP son las siglas de Protocolo de Control de Transmisión/Protocolo Internet. Consiste en un conjunto de reglas que permiten comunicarse a través de una red a los sistemas con componentes de hardware y software distintos. En el presente estudio los sistemas a comunicarse son AS/400 y Windows.

TCP/IP debe ser instalado y configurado en Windows como en AS/400, respectivamente. Para poder utilizar TCP/IP en AS/400, es preciso instalar y configurar la función básica de comunicaciones TCP/IP que es parte de los programas bajo licencia del sistema operativo OS/400 de AS/400.

Antes de la puesta a punto de TCP/IP se debe tener conocimiento previo acerca de:

- El sistema de direcciones TCP/IP. Se debe tener conocimiento de la dirección IP del sistema AS/400, la máscara de subred, y la dirección del siguiente salto, o sea, la dirección del direccionador IP (de haberlo) que direcciona el tráfico de la red dentro y fuera de la organización.
- Las tareas de configuración y administración en AS/400, en este caso la administración y configuración de TCP/IP, de recursos de comunicaciones y de líneas de comunicaciones.
- Información acerca del sistema y de la red a la que dicho sistema se conectará: Tipo de adaptador de comunicaciones, nombre de recurso del puerto de comunicaciones, descripción de línea de comunicaciones, nombre del sistema principal y nombre del dominio para el sistema y dirección IP para el servidor de nombres de dominio.

La puesta a punto de TCP/IP en el sistema AS/400 consta de los siguientes pasos:

1. Identificación del sistema AS/400 en la red local, es decir, la red donde está conectado directamente el AS/400. Esto implica determinar una descripción de línea para el adaptador de comunicaciones de la red y definir una interfaz TCP/IP para dar una dirección IP al AS/400.
2. Definición una ruta TCP/IP. Esto permite al AS/400 comunicarse con sistemas en redes remotas.
3. Definición un nombre de dominio y un nombre de sistema principal locales. Así se asigna un nombre al sistema.
4. Identificación de los nombres de los sistemas de red. Se debe crear una tabla local de sistemas principales.
5. Iniciación de TCP/IP.
6. Verificación del funcionamiento de TCP/IP

De igual modo que se configura TCP/IP en AS/400 se debe hacer la configuración de este protocolo en el sistema Windows.

### 2.2.2 Aplicaciones de TCP/IP

Después de haber instalado y puesto a punto TCP/IP en el AS/400, estas son algunas de las aplicaciones que es posible ejecutar:

Telnet(Red Teletipo): Permite al usuario desde su PC iniciar una sesión en un sistema AS/400.

FTP(Protocolo de Transferencia de archivos): Permite intercambiar archivos entre los dos sistemas.

SMTP(protocolo Simple de Transferencia de Correo): Permite enviar correo electrónico.

Servidor de Correo POP(Protocolo de Oficina Postal): Permite intercambiar correo incluyendo archivos, video, de imagen, audio o binarios, con cualquier cliente que soporte la interfaz de correo POP V3.

LPR(Peticionario de Impresora de líneas): Permite imprimir archivos de AS/400 a impresoras de la red Windows o viceversa.

LPD(Daemon de Impresora de Líneas): Permite que sistemas remotos como un PC, impriman archivos en el sistema AS/400.

### 2.3 INTEGRACIÓN AL NIVEL DE SISTEMA OPERATIVO

Una vez que se tiene la capa de conectividad, es decir, todo el cableado que une las dos plataformas, y un protocolo estándar de transmisión de información que reconocen los dos sistemas se puede hablar de la comunicación entre los dos sistemas operativos como son OS/400 y Windows.

La comunicación entre estos dos sistemas ofrece los siguientes beneficios:

- Gestión del sistema OS/400 de una manera más fácil, a través de la interfaz grafica de Windows.
- Transferencia de archivos entre los dos sistemas.
- Importación, exportación de la base de datos del OS/400, DB2.
- Utilización las aplicaciones de TCP/IP.
- Creación de aplicaciones que puedan acceder a diferentes tipos de información u otras aplicaciones de AS/400.

Estos beneficios los podemos obtener si se instala y configura el software que ofrece IBM para obtener la integración de OS/400 con Windows.

### 2.3.1 Software de IBM para las comunicaciones y gestión de AS/400 con Windows

Entre el software más importante que IBM ofrece para estos fines se encuentran los siguientes:

#### 2.3.1.1 Client Access [DTP004]

IBM AS/400 Client Access es la solución para la conectividad del PC con el AS/400. Gracias al conjunto de funciones que Client Access tiene disponibles, el escritorio del PC puede utilizarse para acceder a los sistemas AS/400 y administrarlos.

Ofrece un potente conjunto de posibilidades de cliente-servidor que permite conectar los PC a los sistemas AS/400. Client Access permite a los usuarios finales y a los programadores de aplicaciones potenciar la información, las aplicaciones y los recursos de la empresa. Las características de la interfaz gráfica de usuario (GUI) multiplican la productividad de los usuarios finales que acceden a los recursos existentes en los sistemas AS/400.

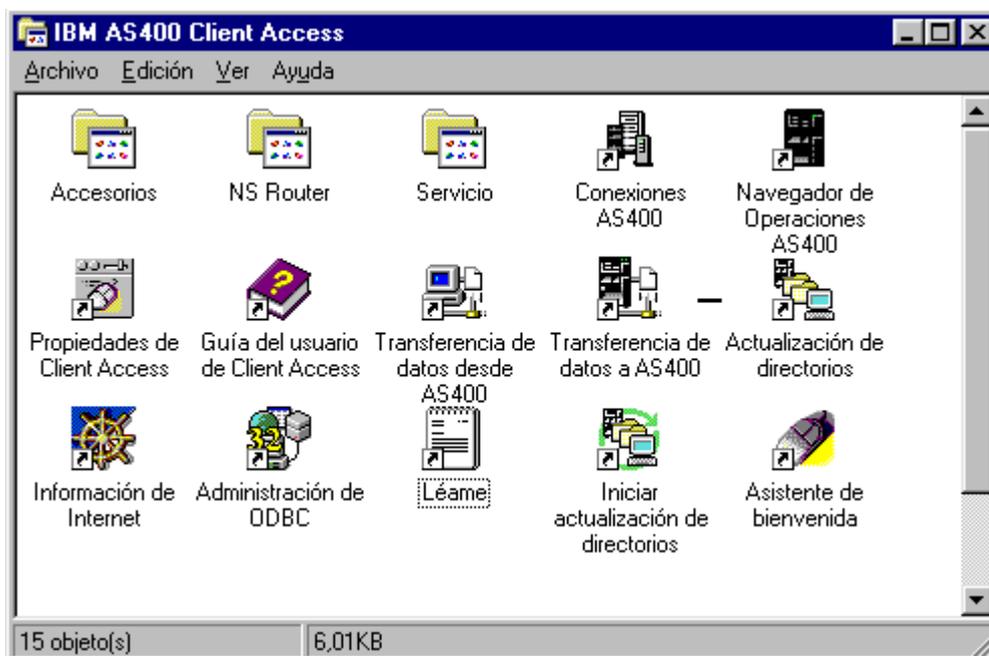


Figura 2.2 Herramientas de Client Access para Windows 95/NT.

Compatible con los sistemas operativos Windows 95, Windows 98 y Windows NT, etc., Client Access presenta una gran variedad de funciones, aplicaciones y habilitadores entre los clientes PC y el AS/400. Algunas de las características que están disponibles permiten:

- Transferir datos entre el cliente PC y el sistema principal AS/400 utilizando la opción Transferencia de datos.
- Sacar partido de la emulación de terminal y de impresora utilizando PC5250.
- Gestionar el sistema AS/400 utilizando la interfaz gráfica de usuario de AS/400 Operations Navigator.
- Utilizar la interfaz ODBC para trabajar con las bases de datos de AS/400.

IBM ofrece una documentación completa acerca de cómo se instala, configura y utiliza Client Access, además, ayuda en línea.

Para administrador Client Access para Windows, se necesita conocer los sistemas operativos de Windows y AS/400. Además, los siguientes temas:

### **Consola PC5250 Client Access para AS/400.**

El soporte de consola de Client Access para AS/400 hace posible que un PC funcione como una consola AS/400. El PC puede conectarse directamente al puerto de comunicaciones de AS/400. El PC que se utiliza como consola se conecta al sistema AS/400 mediante un cable especial de consola. El cable conecta el puerto serie del PC con un adaptador entrada salida que esta instalado en el sistema AS/400. El PC que se utilice como consola de AS/400 debe ejecutar la aplicación PC5250 para poder así emular una estación de pantalla 5250 conectada localmente.

La consola de PC se configura en el sistema AS/400 como un controlador. En el PC se requiere que se instale y configure la aplicación PC5250.

Mediante esta emulación se puede visualizar la interfaz propia del sistema AS/400 en el PC y acceder a todos los menús y comandos del AS/400, dependiendo claro del perfil de usuario y su nivel de autorización.

### **Conexiones AS/400**

Configurar conexiones a sistemas AS/400 utilizando la interfaz Conexiones de Client Access simplifica el modo de crear y gestionar las conexiones. Las funciones de Client Access utilizan las conexiones que están definidas con el programa Conexiones AS/400.

Conexiones AS/400 permiten agrupar definiciones de conexión en grupos de entornos. Un entorno es un grupo de conexiones que tienen algún atributo en común.

La creación y configuración de comunicaciones permite que se cree una interfaz entre el sistema AS/400 y Windows. Esta interfaz es útil al usar otras aplicaciones de Client Access como la emulación PC 5250 o en el caso de transmisión de datos vía ODBC ya que debe haber una Conexión subyacente para obtener los servicios de estas aplicaciones.

### **ID de usuario por omisión e ID de usuario actual**

Cuando se define una conexión, se puede especificar un ID de usuario por omisión. El ID de usuario por omisión se sitúa en el indicador de campo ID de usuario del panel de inicio de sesión del AS/400. Se puede escribir la contraseña asociada con el ID de usuario por omisión o bien especificar un ID de usuario y contraseña distintos. El concepto de un ID de usuario por omisión en contraste con un ID de usuario actual le permite iniciar la sesión en el AS/400 en distintos momentos utilizando distintos ID. Esto puede ser útil si tiene múltiples ID de usuario con distintos privilegios de acceso en el sistema AS/400.

### **Transferencia de datos desde AS/400 al PC**

Esta aplicación permite transferir datos desde un sistema AS/400 al PC. Puede visualizar los datos, almacenarlos en disquete o en un disco duro del PC, o bien imprimirlos. Antes de transferir los datos, debe crear o abrir una petición de transferencia que contenga información acerca de los datos que desea transferir. Para crear una petición de transferencia, debe especificar las opciones siguientes:

#### Opciones AS/400

- Nombre de sistema.
- Nombre de archivo.
- Opciones de datos.

#### Opciones PC

- Dispositivo de salida.
- Nombre de archivo.
- Opciones de formato.

### **Transferencia de datos del PC al AS/400**

Permite transferir datos del PC a una biblioteca/archivo(miembro) en el sistema AS/400. Para crear una petición de transferencia al sistema AS/400, debe especificar las opciones siguientes:

Opciones PC

- Nombre de archivo

Opciones AS/400

- Sistema
- Biblioteca/archivo(miembro)

#### **2.3.1.2 Kit de herramientas de desarrollo de aplicaciones**

El Kit de Herramientas de Desarrollo de Aplicaciones de Client Access contiene información, programas de ejemplo, archivos de cabecera y archivos de biblioteca que permiten a los programadores de aplicaciones crear aplicaciones personalizadas que enlacen con el AS/400 utilizando interfaces Client Access en el PC.

Para utilizar las interfaces Client Access, un programador de aplicaciones debe incluir los archivos de cabecera apropiados en su programa y enlazarlo con la biblioteca CWBAPI.LIB. CWBAPI.LIB que se encuentra en el directorio TOOLKIT\LIB en el PC.

#### **Interfaces de Programación de Aplicaciones (API)**

IBM incluye con Client Access un conjunto de APIs como objetos de tipo OLE 2. Una muestra de estas herramientas se encuentra como archivos de cabecera dentro del directorio TOOLKIT\INCLUDE en el PC una vez instalado Client Access. Algunos de ellos y su uso se detallan en la siguiente tabla:

<b>APIS DE CLIENT ACCESS PARA ENLACE DE APLICACIONES CON AS/400</b>	
<b>Archivo de cabecera</b>	<b>Descripción</b>
CWBCOM.H	Archivo de cabecera API de comunicaciones
CWBDB.H	Estas API permiten que una aplicación acceda a información de base de datos AS/400.
CWBDQ.H	Archivo cabecera API de colas de datos
CWBNP.H	Las APIs de impresión de red permiten que las aplicaciones puedan gestionar la redirección de puertos en el cliente
CWBRS.H	API de mandatos remoto
CWBSOAPI.H	API de acceso a objetos del Sistema, etc.

**Tabla 2.2 APIs De Client Access Para Enlace De Aplicaciones Con AS/400.**

**Archivos MAK**

Los directorios de TOOLKIT\SAMPLES contienen los siguientes archivos MAK. Estos archivos MAK pueden construirse para crear programas de ejemplo de trabajo. Estos programas de ejemplo ilustran la utilización de las distintas API de Client Access. Algunos ejemplos son:

<b>ARCHIVOS MAK DE CLIENT ACCESS</b>	
<b>Archivo MAK</b>	<b>Descripción</b>
LSTJOB.MAK	Programa de ejemplo de acceso a objetos del sistema que ofrece una interfaz de usuario para trabajar con trabajos AS/400.
LSTSPLF.MAK	Programa de ejemplo de acceso a objetos del Sistema que ofrece una interfaz de usuario para trabajar con archivos de spool de AS/400.
SMPLCOM.MAK	Programa de ejemplo de comunicaciones que ilustra como obtener una lista de sistemas AS/400 que están conectados actualmente.
SMPLSQL.MAK	Programa SQL de ejemplo que ilustra algunas de las peticiones usuales de catálogos, base de datos nativa(NDB) y SQL.
SYPROG.MAK	Programa de ejemplo de la API de seguridad que ilustra la verificación del cambio de un ID de usuario y contraseña.
VBPCSAPI.MAK	Programa de PC5250 de ejemplo de Visual Basic.

**Tabla 2.3 Archivos Mak De Client Access.**

### **Controlador ODBC de Client Access**

La Conectividad de Bases de Datos Abiertas (ODBC) es una interfaz definida por Microsoft. ODBC ofrece una interfaz estándar que le permite acceder a distintas bases de datos utilizando el Lenguaje de Consulta Estructurada (SQL). Una aplicación que utilice la interfaz ODBC puede conectarse a cualquier base de datos que disponga de un controlador ODBC, normalmente sin tener que realizar cambios en el código de la aplicación.

El controlador ODBC de Client Access ODBC (32 bits) permite que las aplicaciones accedan a datos de una base de datos de AS/400 mediante la interfaz ODBC.

### **Subsistema QSERVER**

Cada función de Client Access se comunica con un programa que se ejecuta en el AS/400. Este programa se conoce como el programa servidor de sistema principal. El programa servidor de sistema principal para el servicio de bases de datos es QIWS/QZDAINIT. Bajo condiciones normales, este programa se inicia automáticamente y el usuario sólo debe comprobar que el subsistema está ejecutándose.

El programa servidor de bases de datos (QZDAINIT) se ejecuta bajo el subsistema QSERVER o bajo el subsistema de comunicaciones por omisión (QCMN o QBASE, que es el valor por omisión instalado). Si se utiliza TCP/IP u otra pila de comunicaciones proporcionada con Client Access, se estará ejecutando el servidor bajo el subsistema QSERVER.

Debido a los cambios efectuados al modo en que se arrancan los subsistemas en la V3R1, puede que se deba arrancar el subsistema QSERVER manualmente. Para hacerlo, se debe usar el siguiente mandato en un indicador de mandatos AS/400:

```
STRSBS QSERVER
```

Si se experimenta problemas en la conexión con el servidor o si no se ve el subsistema QSERVER en el sistema, tal vez deba comprobar si tiene el producto 5763SS1 opción 12 (Servidores de sistema principal) instalado.

### **Servidores de Sockets del Sistema Principal AS/400**

Para que funcione Client Access se debe ejecutar los servidores sockets del sistema AS/400, utilizando el mandato siguiente:

STRHOSTSVR SERVER(\*ALL)

Para poder utilizar los servidores de sockets, debe estar habilitado el perfil QUSER en el sistema AS/400 sin contraseña. Especificar que no haya ninguna contraseña significa que nadie puede iniciar la sesión en el AS/400 con el perfil QUSER. Además, debe establecer la contraseña a no caducable con un valor temporal para la no-caducidad. Para establecer el perfil QUSER con los valores necesarios, escriba el siguiente mandato en el indicador de mandatos de AS/400:

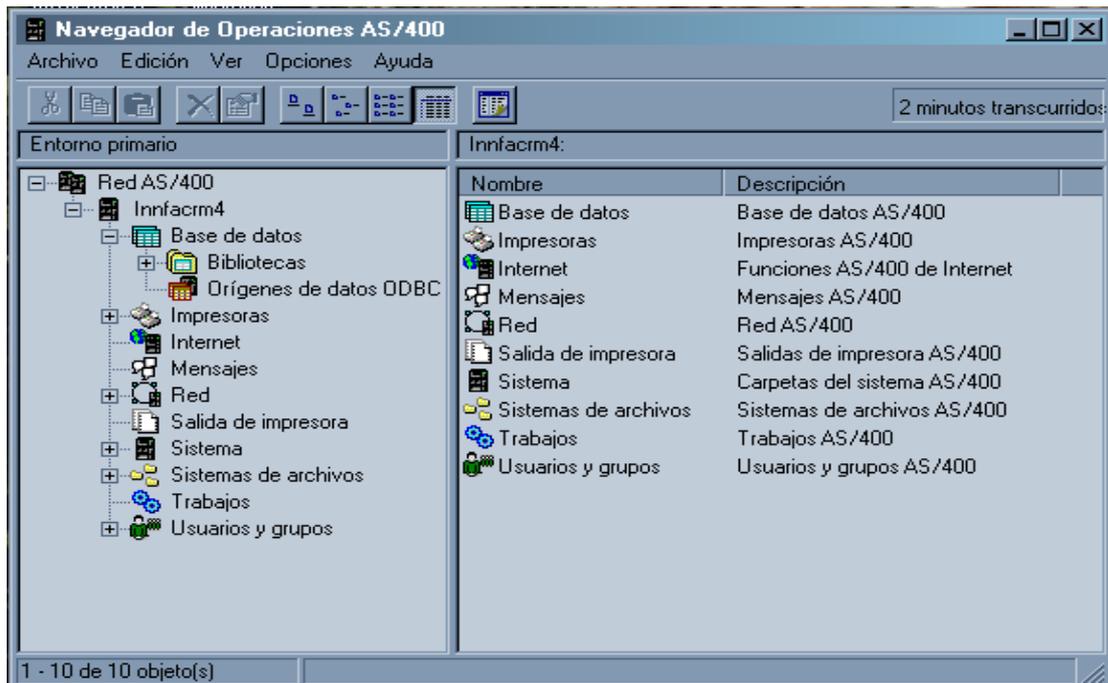
```
CHGUSRPRF USRPRF(QUSER) PASSWORD(*NONE) PWDEXP(*NO) STATUS(*ENABLED)
PWDEXPITV(*NOMAX)
```

### **Impresoras**

Client Access permite imprimir utilizando una sesión de impresora PC5250 o una impresora de red. La sesión de impresora PC5250 le permite utilizar una impresora conectada al PC como si estuviera conectada a un sistema AS/400. Por otro lado, la función de impresora de red le permite utilizar una impresora conectada a un sistema AS/400 como si estuviera conectada al PC.

#### **2.3.1.3 Navegador de operaciones del AS/400**

El navegador de operaciones del AS/400 es una interfaz gráfica para los clientes de Windows 95/NT. Con el Navegador de operaciones del AS/400, se pueden utilizar los conocimientos sobre Windows 95/NT para gestionar y administrar los sistemas AS/400. Se puede trabajar con la administración de bases de datos, sistemas de archivos, administración de la red Internet, usuarios y grupos de usuarios. También se pueden planificar copias de seguridad sistemáticas y visualizar el inventario de hardware y software de AS/400. La Figura muestra un ejemplo de la pantalla. Las utilidades las detallamos a continuación:



**Figura 2.3 Ventana Navegador De Operaciones AS/400.**

Con **Base de datos** se puede administrar los objetos de base de datos DB2/400. Para cada AS/400 al que esté conectado, puede trabajar con Bibliotecas y Orígenes de datos ODBC.

Con **Sistemas de archivos** se consigue visualizar una lista de todos los sistemas de archivos AS/400 sobre los que está autorizado. Puede acceder a objetos AS/400 especificando la vía del objeto.

Mediante **Internet** se puede visualizar una lista de las funciones de Internet a las que puede acceder desde el Navegador de Operaciones AS/400.

Usando **Red** se consigue visualizar los tipos de recursos que puede gestionar para la red AS/400. Estos recursos incluyen protocolos, punto a punto y servidores.

Pulsando **Usuarios y Grupos** se obtiene una lista completa de los usuarios y grupos de un sistema AS/400. Desde allí, puede administrar usuarios y grupos AS/400. Para cada AS/400 al que esté conectado el PC, puede trabajar con Todos los usuarios, Grupos y Usuarios que no forman parte de un grupo.

Pulsando **Copia de seguridad** se puede planificar las copias de seguridad diarias, semanales o mensuales de los datos de usuario. Además, puede especificar una unidad de cintas diferente o un grupo de cintas diferente que debe utilizarse en cada copia de

seguridad planificada. Y, naturalmente, hay un espacio para especificar la hora del día y el día de la semana en que desea que se realice la copia de seguridad planificada.

Con **Hardware y Software** se obtiene una lista de todo el hardware y software del sistema.

## **2.4 NIVEL DE INTEGRACIÓN DE DBMS**

Compartir la información de base de datos entre sistemas diferentes en la actualidad es de vital importancia. Las aplicaciones relevantes en estos momentos de la integración de base de datos son la construcción de Data Warehouse, software de Internet, Sistemas de Información Gerencial, etc.

En esta investigación se ha usado un concepto bastante fácil a los usuarios y a las aplicaciones como es ODBC. Tanto AS/400 como Windows abren sus sistemas de base de datos a este tipo de conectividad.

### **2.4.1 ODBC(Conectividad Abierta de Base de Datos) [WWW003]**

ODBC es una interfaz para base de datos que ha sido definida por Microsoft. La biblioteca ODBC contiene llamadas de funciones que le permiten a las aplicaciones conectarse a los DBMS, ejecutar sentencias SQL y buscar resultados. Esta ventaja existente en Windows se puede utilizar para acceder a bases de datos AS/400 en un ambiente cliente-servidor.

A continuación ampliamos la información ODBC en este ambiente:

#### **2.4.1.1 Sistemas Cliente/Servidor basados en ODBC**

La instalación y uso de un sistema cliente/servidor basado en ODBC, se puede resumir(para la parte cliente) en la lista de pasos que se expone a continuación:

#### **Personalización del sistema ODBC**

##### **1. Instalación de drivers**

Para cada Sistema Gestor de Base de Datos(SGBD) que sea necesario acceder desde el cliente se hace necesario instalar un Driver distinto que se encargue de traducir las llamadas ODBC al lenguaje específico de SGBD.

##### **2. Creación de fuentes de datos(Data Sources)**

Utilizando el programa de mantenimiento o el **ODBC manager** se asocia un Drivers a una localización de un SGBD concreto (base de datos, directorio...),

dando a esta asociación un nombre único en el sistema que identificará de forma conveniente la **fente de datos**. En este mismo paso se determinan los parámetros que regirán el acceso a esta fuente (usuario, password, tiempo de espera, tamaño de buffers...).

### 3. Preparación del entorno

La aplicación llama al ODBC, el cual carga el ODBC manager y prepara el entorno de trabajo.

### 4. Conexión a una fuente de datos

La aplicación determina a que **fente de datos** se ha de conectar y realiza la llamada de conexión oportuna, indicando el nombre identificador de la misma y los valores de los parámetros de desconexión que se quieran establecer (si estos no son indicados se consideraran los valores por defecto dados en la instalación). El **ODBC manager** carga el Driver oportuno que a su vez abre la conexión con el SGBD que corresponda.

### 5. Construcción de la sentencia SQL.

Se puede realizar de varias formas:

- a. Una sentencia fija contenida en una constante de la aplicación.
- b. Una sentencia tecleada por el usuario a través de un formato o interfaz apropiada.
- c. A través de un prototipo y mediante procedimientos de la aplicación que asocia valores de variable a partes de la sentencia en sí.
- d. Con sentencias SQL parametrizables dando valores a los parámetros y realizando el enlace a los mismos.

### 6. Envío de la sentencia SQL al sistema ODBC.

Mediante la función oportuna, la aplicación remite la sentencia SQL(a través de un parámetro de una función de la API) e indicando el Andel de la conexión que ha de recibir la sentencia(obtenido en el paso 5). El ODBC manager, al recibir esta llamada puede proceder a dos acciones:

- a. Preparar y optimizar la sentencia para su posterior ejecución de forma repetitiva.

- b. Ejecutar la sentencia. Al ejecutar la sentencia se retorna un **Handle** identificador de la misma.

## **7. Recorrido del set de datos**

Si la sentencia es de tipo SELECT, u otra que retorne datos se procederá a ir recorriendo las distintas líneas del set retornado, remitiendo las funciones que permitan moverse en las mismas, identificando siempre la sentencia a manejar con su Andel.

Los movimientos a realizar dependerán de SGBD y del Driver y de la propia conexión. En general permitirán los movimientos hacia atrás, hacia adelante, el primero y al último.

## **8. Recogida de resultado**

En cada posición del cursor o línea, pueden obtenerse los contenidos de sus columnas mediante funciones en las que se asocia los nombres o posiciones de las mismas a variables de la aplicación. Estas funciones necesitan identificar la sentencia SQL a tratar mediante el Andel o identificador obtenido en el paso 7. El Driver y ODBC manager, realizan las conversiones oportunas de tipos.

## **9. Datos de gran tamaño.**

Si los datos a recoger son de gran tamaño se tendrán que realizar bucles que permitan su extracción desde el ODBC manager en partes, ya que el tamaño de los buffers asignados suele ser limitado(generalmente 2024).

## **10. Número de Líneas afectadas.**

Si la sentencia es del tipo INSERT, UPDATE, DELETE..., se puede remitir una función que pida al ODBC manager, este procederá a descargar y liberar los Drivers cargados y recursos utilizados para la conexión.

## **11. Cerrar sentencia.**

Si es necesario, e identificándola por su Andel, la sentencia puede ser cerrada en cualquier momento.

## 12. Cerrar la conexión.

Identificándolo con su Handle, la conexión con una determinada fuente de datos puede ser cerrada con una llamada al ODBC manager, este procederá a descargar y liberar los Drivers cargados y recursos utilizados por la conexión.

## 13. Liberar ODBC.

Mediante llamada al ODBC manager se le indica la no-necesidad del mismo, lo cual descargará el ODBC manager, si no esta siendo utilizado por otra aplicación.

Con el ODBC Driver, todo tipo de aplicaciones Windows con capacidad ODBC (procesadores de texto, hojas de cálculo, generadores de informes, etc.) puede acceder un servidor para llevar las operaciones con la base de datos, realizando llamadas ODBC y búsquedas SQL en el servidor.

Para tener estos servicios disponibles no se requiere ningún cambio en la aplicación.

### 2.4.1.2 Creación de orígenes de datos en Windows

En el ambiente Windows, Microsoft creó la tecnología ODBC. ODBC es un almacén que alberga controladores. El almacén sirve para gestionar los controladores, y los controladores son los que saben "hablar" con la base de datos.

En ODBC, lo que se tiene que hacer es una simple tarea, se llama **crear un origen de base de datos**, otros le denominan fuente en lugar de origen. Una **fuentes u origen de datos** consiste en el nombre, el controlador y la base de datos.

**En Windows 95, 98 o NT** se usa el ODBC de 32 bits, pero algunos sistemas conservan un ODBC de 16 bits para las aplicaciones de Windows 3.11. Se encuentra en el **Data Source Administrator** del Open Data Base Connectivity, u ODBC.

**En Windows 2000**, ODBC de 32 bits ha cambiado su nombre por el de Orígenes de datos (ODBC) y se encuentra en Herramientas administrativas, en el Panel de control. Puede usar Orígenes de datos (ODBC) para tener acceso a los datos de una gran variedad de sistemas de administración de bases de datos.



**Figura 2.4 Administrador De Orígenes De Datos ODBC.**

Para abrir Orígenes de datos (ODBC), se selecciona **Inicio**, luego **Configuración** y, a continuación, **Panel de control**. Haciendo doble clic en **Herramientas administrativas** se tiene acceso a **Orígenes de datos (ODBC)**.

Se notará que las primeras solapas se tiene a **DSN de Usuario**, **DSN de sistema** y **DSN de archivo**, en este caso:

**DSN De Usuario, nombre de orígenes de datos para el usuario.** A veces, una máquina es utilizada para más de un usuario, los orígenes de datos declarados aquí son exclusivos del usuario.

**DSN De Sistema, nombre del origen de datos para el sistema.** Todos los usuarios de la máquina tienen acceso a estos orígenes de datos.

**DSN De Archivo, nombre del origen de datos en archivo.** Se crea un archivo con la extensión DSN, que sirve como origen de datos y puede ser distribuido a otros usuarios. Este origen de datos es el que usa Excel por omisión cuando hace consultas.

Esta otra solapa importante que es **ODBC Drivers u Controladores ODBC**. Aquí se ven todos los controladores disponibles en la máquina. De esta forma se puede consultar si se dispone del controlador que necesita y si es la versión conveniente. Regularmente los controladores de base de datos vienen con un programa **SETUP** que los instala.

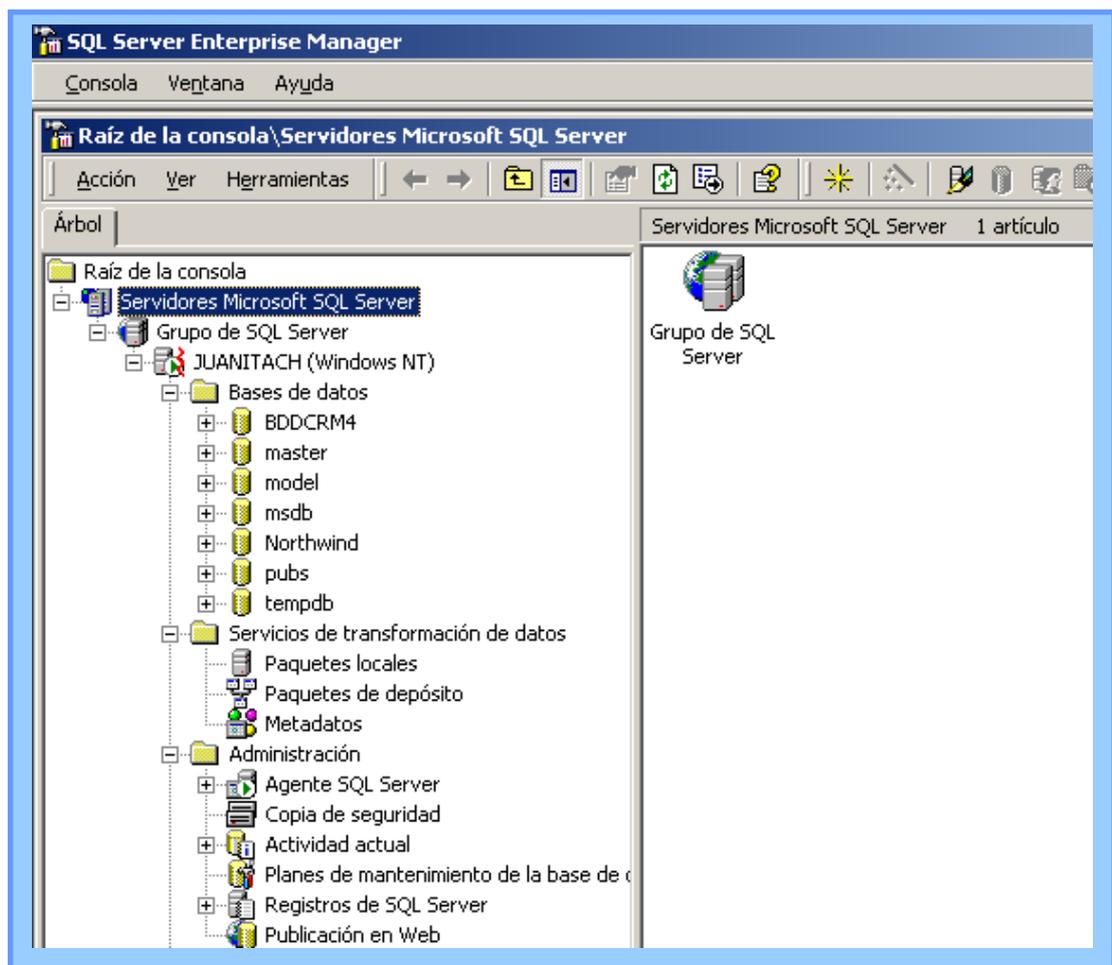
**2.4.2 Integración de DB2 de AS/400 con SQL de Windows NT Server**

DB2 y SQL Server utilizan conexiones ODBC para permitir acceso a sus datos. DB2 por su parte hace uso de conexiones Client Access y el controlador Client Access ODBC de 32 bits y SQL Server utiliza sus herramientas de Importación y Exportación de Datos.

**2.4.2.1 Generalidades de Microsoft SQL Server [DTP005]**

Microsoft SQL Server 2000 es un sistema de gestión de Bases de Datos relacionales cliente-servidor de alto rendimiento.

SQL Server se ejecuta sobre redes basadas en Microsoft Windows NT Server utilizando procesadores Intel o DEC Alpha AXP o compatibles y puede instalarse como sistema de base de datos de escritorio con Windows NT Workstation, Windows 95, Windows 98 o Windows 2000.



**Figura 2.5 Administrador Corporativo De SQL Server.**

#### 2.4.2.1.1 Descripción de los servicios de transformación de datos

Los Servicios de transformación de datos (DTS, *Data Transformation Services*) proporcionan la funcionalidad para importar, exportar y transformar datos entre Microsoft SQL Server y cualquier formato de archivo OLE DB, ODBC o de texto. Al utilizar DTS es posible:

- Generar almacenes y puestos de datos en Microsoft SQL Server al importar y transferir los datos desde varios orígenes heterogéneos interactivamente o automáticamente.
- Crear objetos de transformación personalizada que pueden integrarse con productos de otras compañías.
- Tener acceso a aplicaciones mediante proveedores de OLE DB de otras compañías

DTS también permite:

- Crear transformaciones personalizadas.
- Transferir objetos completos de bases de datos entre orígenes de datos de SQL Server 2000 de origen y de destino. La tarea de transferencia de objetos de SQL Server puede utilizarse para transferir todos los metadatos y los datos de algunos o todos los objetos de una base de datos de SQL Server 2000 a otra base de datos de SQL Server 2000.
- Importar y exportar datos.

Puede **importar y exportar** datos desde Microsoft SQL Server mediante varias herramientas e instrucciones de Transact-SQL. También puede escribir sus propios programas para importar y exportar datos con los modelos de programación y las interfaces de programación de aplicaciones (API) disponibles con SQL Server.

Los métodos para copiar datos a y desde SQL Server incluyen el uso de lo siguiente:

- Los asistentes para importación y exportación con DTS (Servicios de transformación de datos) o el Diseñador DTS para crear paquetes DTS para importar y exportar datos.
- La duplicación de SQL Server para distribuir datos por la empresa.

La tecnología de duplicación de SQL Server le permite crear copias duplicadas de los datos, mover estas copias a ubicaciones diferentes y sincronizar

automáticamente los datos para que todas las copias contengan los mismos valores de datos.

- El programa de símbolo del sistema **bcp** para importar y exportar datos entre SQL Server y archivos de datos.
- La instrucción BULK INSERT para importar información desde archivos de datos a SQL Server.
- La instrucción SELECT INTO para crear nuevas tablas basadas en tablas existentes.

#### **2.4.2.1.2 Asistentes para importación y exportación con los servicios de transformación de datos**

Los asistentes para importación y exportación con los Servicios de transformación de datos (DTS) permiten al usuario crear interactivamente paquetes DTS que pueden utilizarse para importar, exportar, validar y transformar datos heterogéneos mediante OLE DB y ODBC. Los DTS también pueden copiar esquemas y datos entre bases de datos Relacionales. Si se utiliza la capacidad para transferir objetos de DTS, también se puede copiar índices, procedimientos almacenados, restricciones de integridad referencial y otros objetos de la base de datos entre equipos que ejecutan Microsoft SQL Server versión 2000.

Los DTS comprueban si la tabla de destino existe y, si se desea, da la opción de quitarla y crearla de nuevo. Si los asistentes para importación y exportación con DTS no crean de manera adecuada la tabla de destino, comprueban que sean correctas las asignaciones de columnas, seleccionan una asignación de tipo de datos diferente o crean la tabla manualmente. Y, a continuación, copian los datos.

Cada base de datos define sus propios tipos de datos y convenciones de nomenclatura de objetos y columnas. Los DTS intentan definir lo mejor posible las asignaciones de los tipos de datos entre un origen y un destino. Sin embargo, se puede sobrescribir las asignaciones de los DTS y especificar propiedades diferentes acerca de tipos de datos, tamaño, precisión y escala del destino en la ficha **Asignaciones de columnas** del cuadro de diálogo **Asignaciones y transformaciones de columnas** si crea una tabla nueva o si quita una tabla existente y crea una nueva.

Los DTS utilizan de manera predeterminada los nombres de los objetos del origen. Sin embargo, puede agregar comillas dobles o paréntesis alrededor de los nombres de una tabla y nombres de columnas que contengan espacios, si lo admite el DBMS.

Los asistentes para la importación y la exportación con DTS permiten al usuario:

- Especificar cualquier configuración personalizada que admita el proveedor OLE DB utilizada para conectarse al origen o al destino de datos.
- Copiar tablas enteras o los resultados de una consulta SQL, como las que implican combinaciones de varias tablas o, incluso, consultas distribuidas.
- Generar una consulta mediante el Generador de consultas dentro del asistente. Esto permite a los usuarios sin experiencia en el lenguaje SQL generar consultas interactivamente.
- Cambiar el nombre, tipo de datos, tamaño, precisión, escala y posibilidad de aceptar el valor NULL cuando se copie el origen al destino.
- Especificar reglas de transformación que gobiernan la forma en que se copian los datos entre las columnas de tipos de datos, tamaño, precisión, escala y posibilidad de aceptar el valor NULL distintos.
- Transferir objetos de base de datos tales como usuarios, funciones, vistas y procedimientos almacenados entre equipos que ejecutan SQL Server 2000
- Guardar el paquete DTS en la base de datos **msdn** de SQL Server, en Microsoft Repository o en un archivo de almacenamiento con estructura COM.
- Especificar la tabla de destino que se utilizará para la duplicación.
- Programar el paquete DTS para su posterior ejecución.

#### **2.4.2.1.3 El Diseñador DTS**

El Diseñador de los Servicios de transformación de datos (DTS) facilita la importación, exportación y transformación de datos heterogéneos. El Diseñador DTS proporciona:

- Un motor de flujo de trabajo orientado a transacciones que le permite especificar una secuencia compleja de operaciones que se realizan como parte de un paquete DTS.

- Movimiento de datos integrado. El Diseñador DTS utiliza una arquitectura OLE DB al 100% para copiar, validar y transformar datos de orígenes de datos cliente-servidor y de los equipos de escritorio más utilizados. Se proporciona conectividad para orígenes de datos de Microsoft Access, Microsoft Excel, Microsoft FoxPro, Microsoft SQL Server, dBase, Paradox, Oracle y AS/400.
- Un método estándar de la industria para compartir información de metadatos y del linaje de datos a través de Microsoft Repository.
- La capacidad de guardar paquetes DTS en Microsoft Repository, en SQL Server o en archivos. Una vez que se ha creado y guardado un paquete, su ejecución puede programarse mediante el Agente SQL Server.

### **Configuración de los componentes de paquetes DTS**

Para construir un paquete, es necesario configurar las conexiones de los orígenes y destinos de los datos, definir las asignaciones de transformación, configurar las propiedades de las tareas y establecer las relaciones de precedencia.

#### **2.4.2.2 Procedimiento de acceso a DB2 AS/400 mediante SQL Server**

Para tener acceso a los datos de DB2 se puede utilizar las funciones y herramientas que brinda Client Access. La herramienta más importante es la del controlador ODBC de 32 bits que se instala en el momento de instalar Client Access en el PC.

El acceso a los datos de DB2 desde SQL Server se puede dividir en los siguientes pasos:

- Conocimiento de la definición de las tablas de DB2 en el AS/400.
- Configuración de las conexiones de origen y destino de los datos.
- Utilización de los Servicios de Transformación de datos de SQL Server.

##### **2.4.2.2.1 Conocimiento de la definición de las tablas de DB2 en el AS/400**

Conocer los nombres de las tablas o tabla que se va a transferir o consultar es indispensable. Si se va a transferir la tabla de modo completo entonces no es necesario saber el nombre de los campos de la o las tablas a transferir. SQL Server permite hacer la transferencia de los resultados de una consulta con sentencias SQL. La consulta puede involucrar a una o varias tablas de DB2, además, de uno o varios campos de las tablas. En la consulta pueden existir parámetros de selección de registros.

#### 2.4.2.2.2 Configuración de las conexiones de origen y destino de los datos

Tener acceso a los datos y tablas de DB2 implica que se debe tener un modelo de comunicación en perfectas condiciones entre AS/400 y el PC.

Dentro de este modelo de comunicación se debe cumplir con los siguientes requisitos:

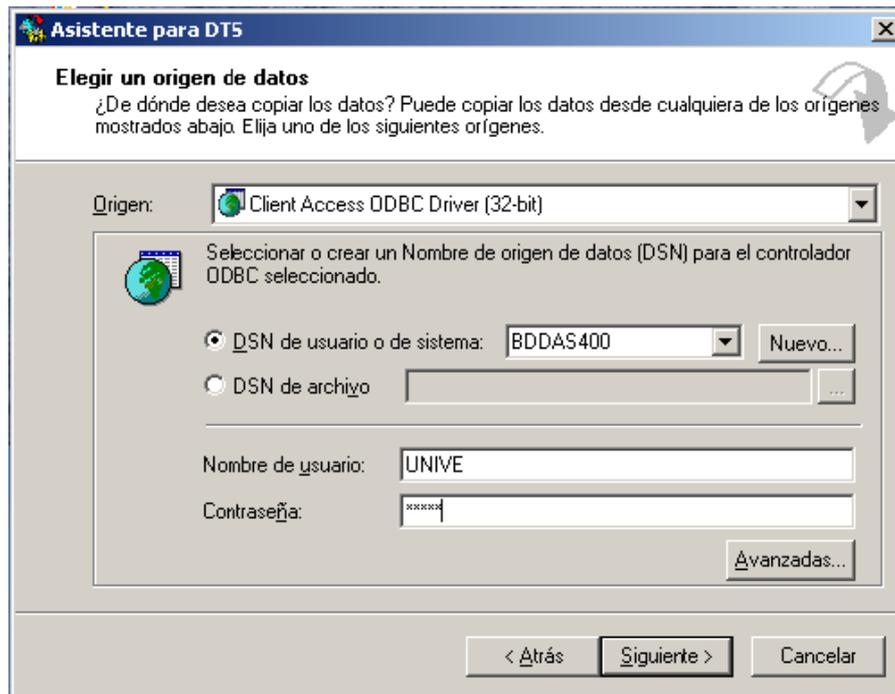
- Poseer entorno de transmisión de datos, por ejemplo TCP/IP entre los equipos. Si es TCP/IP se puede verificar si existe la conexión mediante el comando Ping.
- Disponer de una conexión de comunicación Client Access. La conexión se realiza mediante la herramienta Conexiones AS/400 del menú IBM Client Access. En la configuración de la conexión se ingresan el nombre del sistema, el ID. de un usuario autorizado y la dirección IP del sistema AS/400 que servirán en el momento de la conexión.
- Configuración de un origen de datos ODBC Client Access de AS/400 mediante el administrador de ODBC de Windows.
- Instalación y Configuración de SQL Server en el PC destino. No se debe olvidar de iniciar los servicios de SQL Server.
- Seleccionar la base de datos destino de SQL Server; se puede crear una base de datos nueva o seleccionar una ya existente. Aquí residirán los datos de la transferencia. La definición de la o las tablas destino se puede(n) crear antes o en el momento de la transferencia.. Se puede escoger el mismo nombre de la tabla origen o un nombre nuevo.

#### 2.4.2.2.3 Utilización de los servicios de transformación de datos de SQL Server

A la herramienta que brinda SQL Server para la importación y exportación de datos se puede acceder de algunas maneras:

- En el menú SQL Server de *Programas* del PC en el submenú *Importar y Exportar datos*.
- En el Administrador Corporativo SQL Server en el menú *Herramientas* en el submenú *Servicios de Transformación de datos*.

Independiente del modo de acceso luego de su ejecución, se visualiza una pantalla similar a la de la figura siguiente:



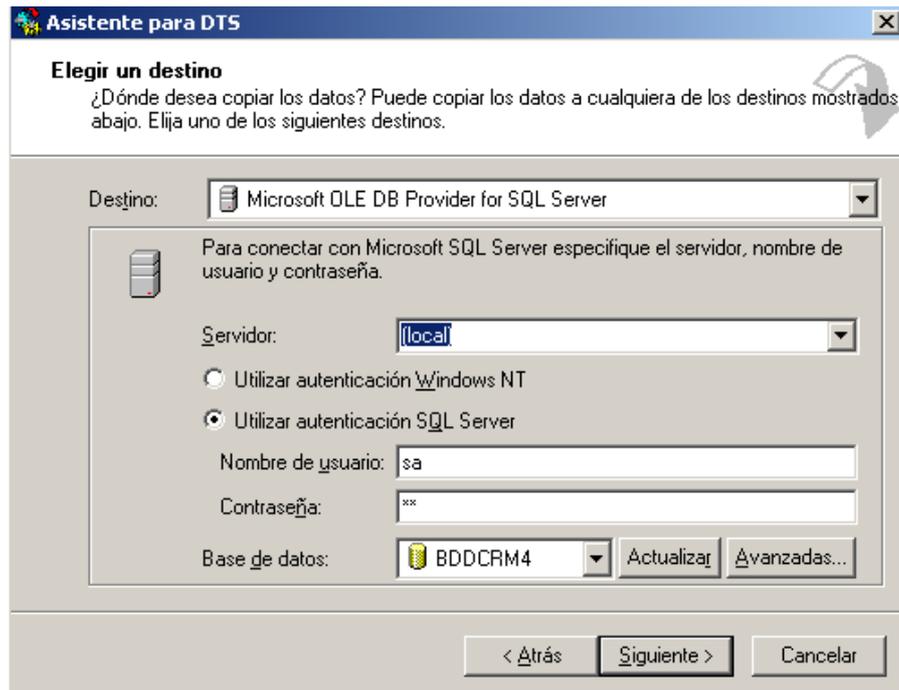
**Figura 2.6 Asistente Para DTS- Origen De Datos.**

En esta ventana se deben ingresar los siguientes datos:

- El tipo de origen de datos: en este caso es Client Access ODBC de 32 bits.
- El nombre del origen de datos(DSN). En este caso debe ser un origen de datos que este direccionado a la base de datos DB2 del sistema AS/400. Véase Anexo de PROCEDIMIENTO DE ACCESO A DB2 AS/400 MEDIANTE SQL SERVER.
- El nombre de usuario que dispone autorización en el sistema AS/400 para poder hacer este tipo de operación.
- La contraseña del usuario.

Una vez ingresado los datos y al pulsar siguiente, el asistente determinará si la información proporcionada es correcta tratando de activar una Conexión Client Access.

Si el asistente logra establecer la conexión pasará al siguiente paso, visualizando la siguiente ventana:



**Figura 2.7 Asistente Para DTS- Destino De Datos.**

En esta ventana se debe ingresar la información siguiente:

- El destino de los datos: SQL Server.
- El nombre del servidor SQL Server.
- El tipo de autenticación de usuario de SQL Server.
- El nombre de usuario que dispone autorización en SQL Server para poder hacer este tipo de operación.
- La contraseña del usuario.
- El nombre de la base de datos en SQL Server.

La siguiente ventana permite seleccionar al usuario si desea hacer la transferencia de una tabla completa o de los resultados de una subconsulta SQL de la tabla.

Si se elige transferir resultados de una subconsulta, se presentara una ventana para el ingreso y análisis de la consulta.

El siguiente paso permite establecer si la consulta se realiza inmediatamente o si solamente se crea un paquete que ejecuta estas tareas. Si el usuario decide crear un paquete DTS también le pedirá que ingrese el nombre del paquete y el destino del mismo.

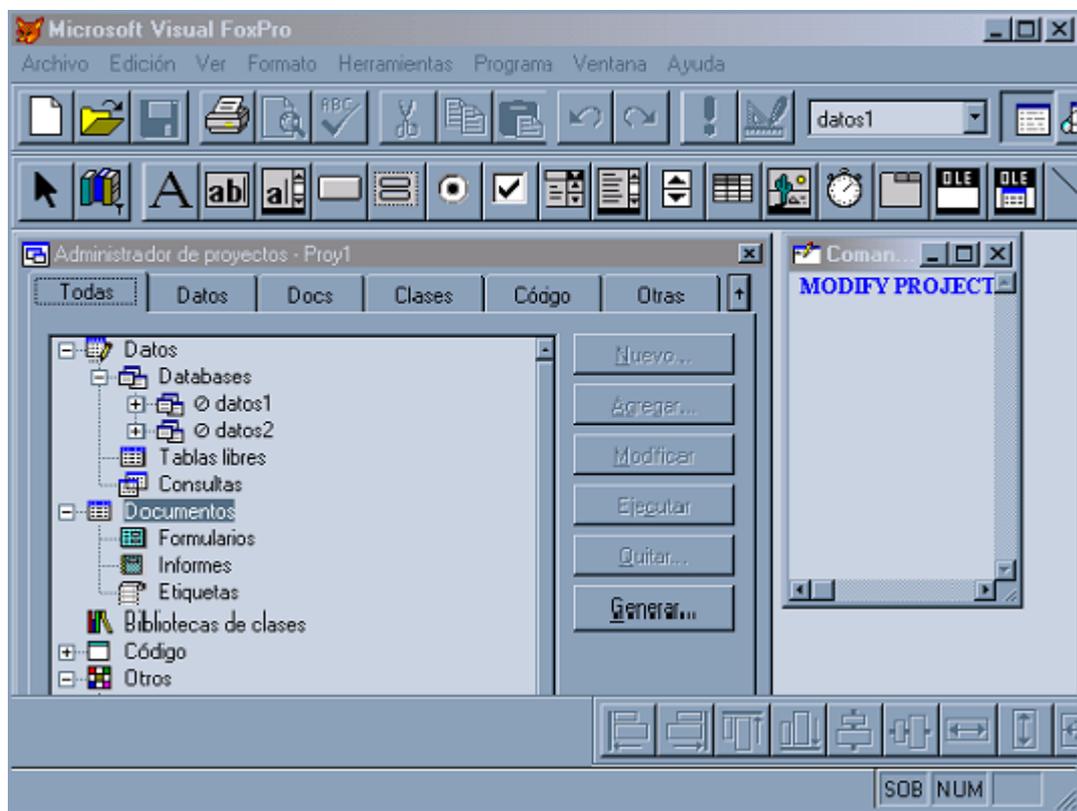
Pulsando Finalizar se da inicio a la transferencia de los datos de un origen DB2 a un destino SQL Server.

Una vez hecha la transferencia de datos, los datos pasan a ser parte del entorno SQL Server y pueden ser manejados según las necesidades del usuario.

### 2.4.3 Acceso a DB2 AS/400 mediante Visual FoxPro

La migración de datos de DB2 a Visual FoxPro también se lo puede realizar mediante ODBC. Para DB2 las herramientas que se utilizan son las mismas que para acceder desde SQL Server, mientras que por parte de Visual FoxPro que se detalla a continuación.

#### 2.4.3.1 Introducción a Visual FoxPro



**Figura 2.8 Interfaz Principal De Visual FoxPro.**

Visual FoxPro es uno de los gestores de Bases de Datos Relacionales del mercado que dispone un completo entorno de desarrollo totalmente orientado al objeto y altamente integrado al sistema Windows. Al igual que otras aplicaciones Windows soporta interfaces MDI (Interfaz de Documentos Múltiples), barras de herramientas, ayuda sensible al contexto, acceso a bibliotecas dinámicas, etc.

### **2.4.3.1.1 El Entorno de desarrollo**

La interfaz de desarrollo de Visual FoxPro dispone del siguiente conjunto de herramientas para el diseño rápido y visual de las aplicaciones.

#### **Administrador de proyectos**

Visual FoxPro incorpora un gestor de proyectos que centraliza la gestión de todos los archivos manejados por una aplicación.

#### **Diseñador de base de datos**

Como apoyo al sistema se ha incorporado un gestor de base de datos. Este gestor maneja todo lo relacionado a tablas, índices, integridad referencial y más información de base de datos.

#### **Diseñador de Formularios**

Totalmente orientado al objeto, permite diseñar el aspecto gráfico final de una aplicación. Permite la inclusión de objetos estándar de Windows en los formularios, tales como casillas de verificación, grupo de opciones, imágenes, etc.

Gestiona de forma transparente al usuario el entorno de datos asociado para cada formulario. En este entorno es posible incluir cualquier tabla, vista local o remota definida en una base de datos.

#### **Diseñador de Informes y Etiquetas**

Visual FoxPro incorpora un potente diseñador de informes y etiquetas para visualizar datos ya sea en pantalla o impresora.

#### **Diseñador de Menús**

Los diseñadores de menús se utilizan para crear menús personalizados en la aplicación.

#### **Asistentes**

Los asistentes pueden ayudar a crear nuevos tipos de ficheros de forma sencilla y rápida. Son guías a la hora de crear controles.

#### **Diseñador de Consultas y Vistas**

Los generadores de vistas y consultas constituyen un apoyo indispensable a la hora de recuperar información mediante expresiones SQL.

### 2.4.3.1.2 Aspectos de base de datos de Visual FoxPro

Se considera una base de datos como el contenedor de información relativa a tablas, relaciones, vistas locales, vistas remotas, conexiones ODBC y procedimientos almacenados:

DATO	DESCRIPCIÓN
Tabla	Guarda el nombre, la descripción y las reglas de validación al nivel de fila.
Vistas	Consultas actualizables realizadas a tablas de la propia base de datos. Se guarda información acerca de las columnas seleccionadas, el criterio de recuperación de registros, el orden de las filas o los campos actualizables.
Campos	Guarda el nombre, la descripción y las reglas de validación al nivel de columna.
Índices	Guarda el tipo de índice: principal, candidato, regular o normal.
Vistas Remotas	Consultas actualizables realizadas a tablas remotas situadas en algún servidor de datos. Se guarda información acerca de las columnas seleccionadas, los criterios de recuperación de registros, el orden de las filas o los campos actualizables.
Conexiones	Contiene información acerca de una conexión hecha vía ODBC a algún servidor de datos. Esta información se utiliza posteriormente para generar vistas remotas.
Procedimientos	Proveen la posibilidad de incluir código que pueda ser utilizado desde el momento en que la base de datos se abre.

**Tabla 2.4 Aspectos De Base De Datos De Visual FoxPro.**

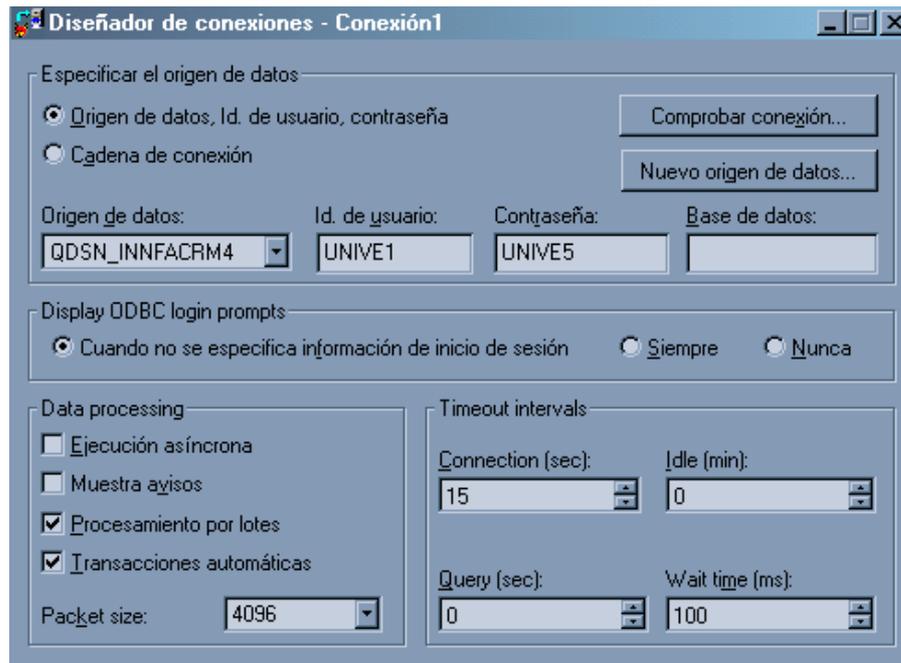
### 2.4.3.1.3 Configuración de Conexiones ODBC en Visual FoxPro

La configuración de una conexión ODBC en Visual FoxPro es sumamente fácil:

En el *Administrador de proyectos* de Visual FoxPro seleccionando en la pestaña *Datos* en el mismo eligiendo *Conexiones* y pulsando el botón **Nueva**, se obtiene la ventana de Configuración de conexiones de Visual FoxPro.

En dicha ventana se puede crear un nuevo tipo de conexión; seleccionando los parámetros deseados para establecer la misma, según el criterio del usuario:

- Se puede configurar la conexión con un nombre de **origen de datos**, el **ID de usuario y contraseña** del sistema remoto o mediante una cadena de conexión. En la lista desplegable *Origen de Datos* de la ventana se visualiza todos los orígenes de datos existentes hasta ese momento en Windows y pueden ser de diferentes tipos como Excel, SQL Server, Access, etc.



**Figura 2.9 Diseñador De Conexiones De Visual FoxPro.**

- Crear un nuevo origen de datos con el botón *Nuevo origen de datos* El Diseñador de conexiones funciona perfectamente con el Administrador de ODBC de Windows para facilitar la creación de conexiones. Pulsando el botón *Nuevo origen de datos* se visualizará el Administrador de ODBC de Windows si se quiere crear un nuevo origen de datos o modificar alguno ya existente.
- Se puede especificar si la información de inicio de sesión como nombre de usuario y contraseña se solicitarán *Siempre o Nunca* cuando se empiece la conexión.
- Especificar parámetros de tipo y velocidad de transmisión.

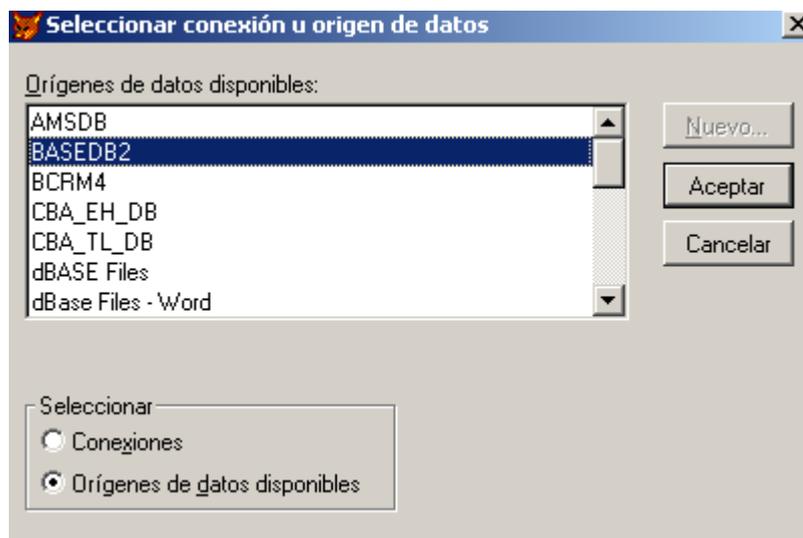
Una vez terminada la configuración de la conexión, se la archivará para luego usarla al crear y ejecutar una vista remota. La definición de dicha conexión se puede mirar en el Administrador de Proyectos una vez archivada.

#### **2.4.3.2 Acceso a DB2 mediante conexiones de ODBC y vistas remotas**

Como se ha mencionado, aparte del acceso a tablas de formato nativo de Visual FoxPro, se puede acceder virtualmente a cualquier fuente de datos cuyo acceso se pueda hacer a través de ODBC utilizando SQL como lenguaje de consulta.

Para acceder a los datos de DB2 podemos usar esta característica que ofrece Visual FoxPro mediante la creación de Conexiones ODBC y Vistas Remotas mediante el siguiente proceso:

1. Con el Administrador de ODBC de Windows se debe configurar un ODBC Client Access con todos los parámetros necesarios para tener acceso a DB2.
2. En el *Administrador de proyectos* de Visual FoxPro seleccionando *Conexiones* se debe configurar una Conexión utilizando el nombre del origen de datos de Client Access configurado para acceso a DB2 en el paso anterior, un ID de usuario valido del sistema AS/400 y la contraseña del mismo y los parámetros de transmisión deseados.
3. Luego de tener la conexión correctamente configurada, se procede a crea una **vista remota**, el servidor remoto debe ser el sistema AS/400. La Vista se realiza de la siguiente manera:
  - Seleccionando en el Administrador de Proyectos en la pestaña Datos la opción *Vistas Remotas* y pulsando *Nueva*, se obtiene una pantalla similar a la del gráfico siguiente. En esta ventana se elige la opción Orígenes de datos disponibles, esto presupone que ya se tiene la conexión ODBC hecha anteriormente, y se elige el origen de datos correspondiente a DB2.



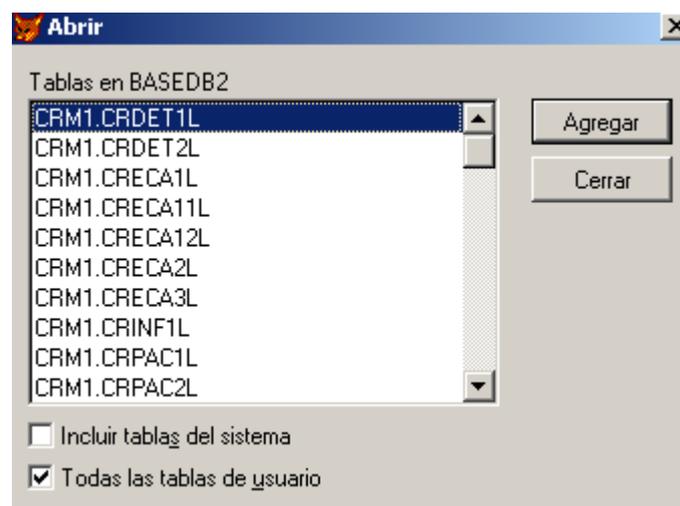
**Figura 2.10 Selección De Origen De Datos Para Visual fox Pro.**

- Al pulsar Aceptar en el paso anterior, se tendrá la ventana de inicio de sesión para el sistema AS/400. Se procede a ingresar los datos y pulsar Enter.



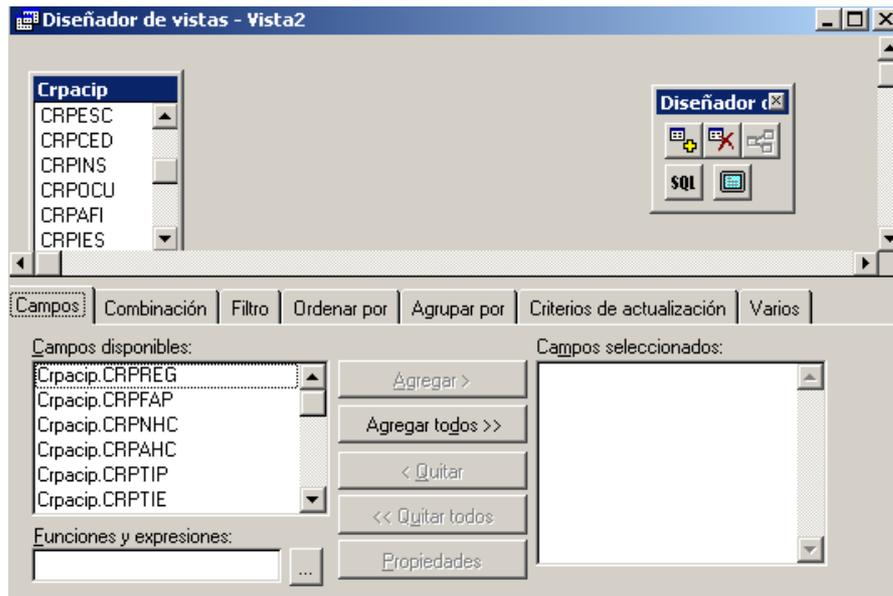
**Figura 2.11 Inicio De Sesión AS/400.**

- Luego de esto se establece la conexión entre DB2 y Visual Fox Pro y se pueden mirar todas las tablas que dispone la base de datos DB2.



**Figura 2.12 Tablas De La Conexión Con DB2.**

- Seleccionando la tabla de donde se desea el conjunto de resultados y pulsamos Agregar se tiene a la vista todos los campos de la tabla.



**Figura 2.13 Diseñador De Vistas.**

- En ese momento se debe seleccionar todos los parámetros de la sentencia SQL en este diseñador de la vista remota y ejecutar la consulta.
- Luego proceder a salir del diseñador y Grabar la vista de datos remotos. El nombre de la vista se verá luego en el Administrador de proyectos para su posterior operación al criterio del usuario o de las aplicaciones.

## **2.5 INTEGRACIÓN DE AS/400 CON HERRAMIENTAS DE DESARROLLO WINDOWS**

En este segmento se analizará dos tipos de herramientas de desarrollo en su manera de integrarse con AS/400. Visual Basic y Delphi son herramientas bastante importantes para los analistas. Estas dos herramientas se basan en: programación orientada a objetos, interfaces MDI, trabajan mediante conjunto de componentes gráficos, pudiendo desarrollar aplicaciones cliente-servidor o monousuario, etc.

El aspecto más importante que se ha determinado en su relación con el AS/400 es aquella que permite acceder, extraer o consultar datos existentes en este sistema. La interfaz que utilizan estos sistemas para su comunicación es ODBC, tanto desde Delphi como desde Visual Basic.

Esta característica que se pone a disposición de los desarrolladores puede ser utilizada para la creación de cualquier tipo de front-end de gestión de datos que se encuentren en AS/400. Aunque no es recomendable la creación de SIG con datos directos de bases de

datos operacionales de AS/400 pues eso iría en contra de las reglas de creación de este tipo de sistemas. Este tema se ampliará más adelante.

Al igual que se hace el acceso por estas herramientas se puede acceder con otras herramientas Windows que acepten el estándar de conectividad de base de datos ODBC, siguiendo similares criterios.

A continuación presentamos el detalle del acceso a los datos de AS/400 mediante cada una de estas herramientas:

### **2.5.1 Acceso a datos de AS/400 desde Visual Basic**

Para acceder a DB2 de AS/400 mediante Visual Basic se dispone de algunos métodos. Se puede utilizar una herramienta existente en la barra de menús denominada **Visdata** o a través de un control denominado RemoteData existente en el grupo de controles de Visual Basic o mediante un control de datos ADO(ActiveX Data Object). Todos estos métodos están disponibles en la versión 6.0 de Visual Basic Profesional.

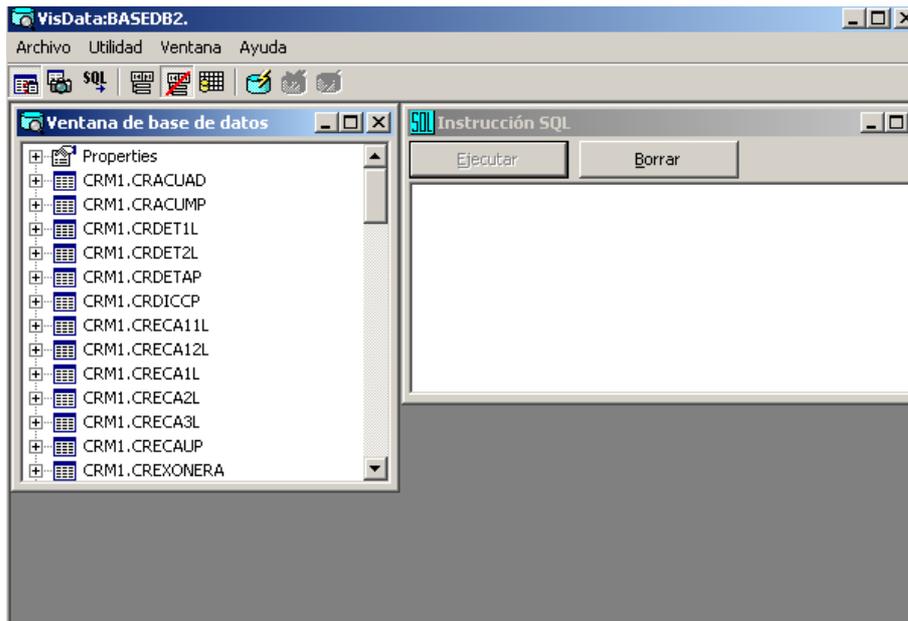
#### **2.5.1.1 Administrador Visual De Datos (VisData)**

VisData es una aplicación creada con Visual Basic. En las ediciones profesional y empresarial, el código fuente de VisData se encuentra disponible en el directorio de ejemplos.

Para tener acceso a VisData, se elige el comando *Administrador visual de datos* en el menú **Complementos**.

**Importar y Exportar (Comando, Menú Archivo de VisData):** Permite importar tablas de otras bases de datos o exportar tablas y resultados de consultas SQL a otra base de datos. Presenta las siguientes opciones:

- **Importar:** Carga los subsiguientes cuadros de diálogo en los que puede elegir el tipo de datos y la tabla que puede importar de otra base de datos.
- **Exportar Tablas:** Exporta todas las tablas seleccionadas en el cuadro de tablas.
- **Resultados De Exportar SQL:** Exporta los resultados de la consulta SQL actual de la ventana Instrucción SQL.
- **Lista:** Enumera las tablas disponibles.



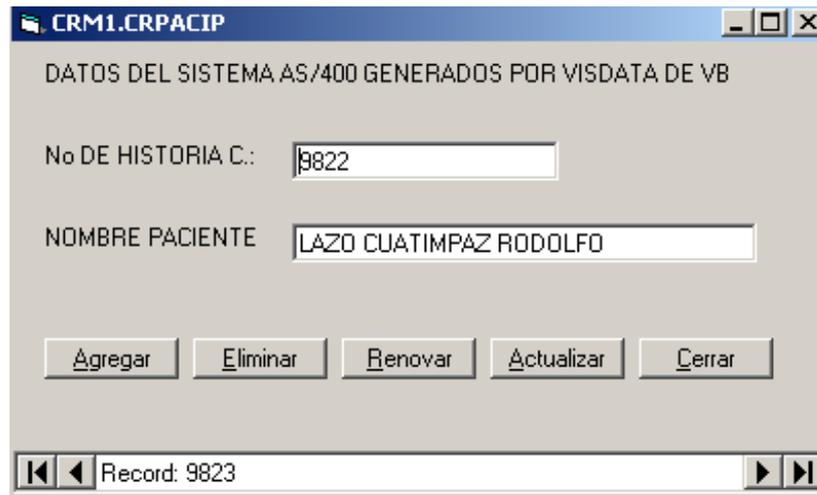
**Figura 2. 14 Administrador Visual De Datos.**

Una vez abierta una base de datos en *Archivo* de **VisData**, utilizando el menú *Utilidad* puede crearse un formulario que acceda a los datos ODBC usando el generador de formularios de **VisData** en el que puede incluir los siguientes datos: nombre del formulario, origen de datos y los campos que se desee incluir en el formulario.



**Figura 2.15 Diseñador De Formulario de Datos.**

El formulario generado se visualiza similar a la siguiente ventana. Es de esta manera que se pueden tener todos los formularios deseados en una aplicación accedendo a datos de DB2 de AS/400.



**Figura 2. 16 Formulario Generado Por Visdata De Visual Basic.**

### 2.5.1.2 Uso del Control Remote Data

El control Remote Data implementa el acceso a los datos usando Remote Data Objects de Microsoft. Esta tecnología le ofrece el acceso sin problemas a muchos formatos de bases de datos y permite crear aplicaciones de reconocimiento de datos sin necesidad de escribir código. El control Remote Data es apropiado para bases de datos grandes de tipo cliente-servidor, incluyendo bases de datos ODBC como Microsoft SQL Server, Oracle, DB2, etc.

El control **Data**, el control **Remote Data** y el control de datos **ADO** de Visual Basic son conceptualmente similares: los tres son *controles de datos* que conectan un origen de datos a un control de enlace de datos.

Para crear una aplicación para bases de datos con el control Remote Data se siguen los siguientes pasos:

1. Dibujar un control Remote Data en un formulario.
2. Si el control Remote Data no está en el Cuadro de herramientas, presionando las teclas Ctrl-T se mostrará el cuadro de diálogo Componentes. En el cuadro de diálogo Componentes, seleccionando Microsoft RemoteData Control. Haciendo clic en *Aceptar* para agregarlo al Cuadro de herramientas.
3. Haciendo clic en Remote Data control para seleccionarlo y presionando F4 se verá la ventana Propiedades de este control.

4. En la ventana *Propiedades*, estableciendo la propiedad *DataSourceName* al DSN de la base de datos con la que quiere conectar, en este caso el ODBC Client Access que permite acceder al sistema AS/400.
5. En la ventana *Propiedades*, haciendo clic en la propiedad *SQL* para introducir una instrucción SQL, por ejemplo:
 

```
SELECT crpnom, crpnhc FROM crpacip WHERE crpnhc > 8000
```
6. Dibujar un control *TextBox* en el formulario.
7. Haciendo clic en el control *TextBox* para seleccionarlo y, en la ventana *Propiedades*, establecer la propiedad *DataSource* al control *Remote Data*.
8. En la ventana *Propiedades*, estableciendo la propiedad *DataField* al nombre del campo de la base de datos que quiere ver o modificar.
9. Repetir los pasos 6, 7 y 8 para cada campo adicional al que quiera tener acceso.
10. Presionar *F5* para ejecutar la aplicación y ver los datos.



**Figura 2. 17 Formulario De Ejemplo Utilizando Control Remote Data De Visual Basic.**

Al igual que con un Text Box se puede utilizar otros controles como un Data List, para ver los datos de DB2 siguiendo un procedimiento anterior.

### 2.5.1.3 Control De Datos ADO

El control de datos **ADO** usa Objetos de Datos ActiveX (ADO) para crear de una manera rápida conexiones entre controles de enlace de datos y proveedores de datos. Los controles de enlace de datos son aquéllos que ofrecen una propiedad *DataSource*. Proveedor de datos puede ser cualquier origen escrito para la especificación OLE DB.

También puede crear fácilmente su propio proveedor de datos mediante el módulo de clase de Visual Basic.

Aunque se puede usar los objetos de datos ActiveX directamente en sus aplicaciones, el control de datos **ADO** tiene la ventaja de que es un control gráfico (con botones *Adelante* y *Atrás*) y una interfaz fácil de usar que le permite crear aplicaciones para bases de datos con un mínimo de código.

Algunos de los controles que se encuentran en el Cuadro de herramientas de Visual Basic pueden ser de enlace de datos, incluidos los controles CheckBox, ComboBox, Image, Label, ListBox, PictureBox y TextBox. De manera adicional, Visual Basic incluye distintos controles ActiveX de enlace de datos, como los controles DataGrid, DataCombo, Chart y DataList. También se pueden crear sus propios controles ActiveX de enlace de datos.

Entre los usos posibles que brindan los objetos de datos ADO se puede mencionar los siguientes:

- Conectar con una base de datos local o remota.
- Abrir una tabla especificada de una base de datos o definir un conjunto de registros basados en una consulta con el Lenguaje Estructurado de Consultas (SQL) o procedimiento almacenado o vista de las tablas de esa base de datos.
- Pasar valores de un campo de datos a controles de enlace de datos, en los que pueda mostrar o cambiar los valores.
- Agregar nuevos registros o actualizar una base de datos según los cambios hechos en los datos que aparecen en los controles enlazados.

Para crear una aplicación para bases de datos cliente o front-end, agregue el control de datos **ADO** a sus formularios del mismo modo en que lo haría con cualquier otro control de Visual Basic. Puede tener tantos controles de datos **ADO** en su formulario como sea necesario.

#### 2.5.1.4 Control DataGrid

El control **DataGrid** es un control vinculado de tipo hoja de cálculo que muestra una serie de filas y columnas que representan registros y campos de un objeto Recordset. Puede usar DataGrid para crear una aplicación que permita al usuario leer y escribir en la mayoría de las bases de datos. El control DataGrid se puede configurar rápidamente en tiempo de diseño con poco o sin ningún código. En tiempo de ejecución, **DataSource** se

puede cambiar mediante programación para ver una tabla diferente, o se puede modificar la consulta de la base de datos actual para que devuelva un conjunto de registros diferente.

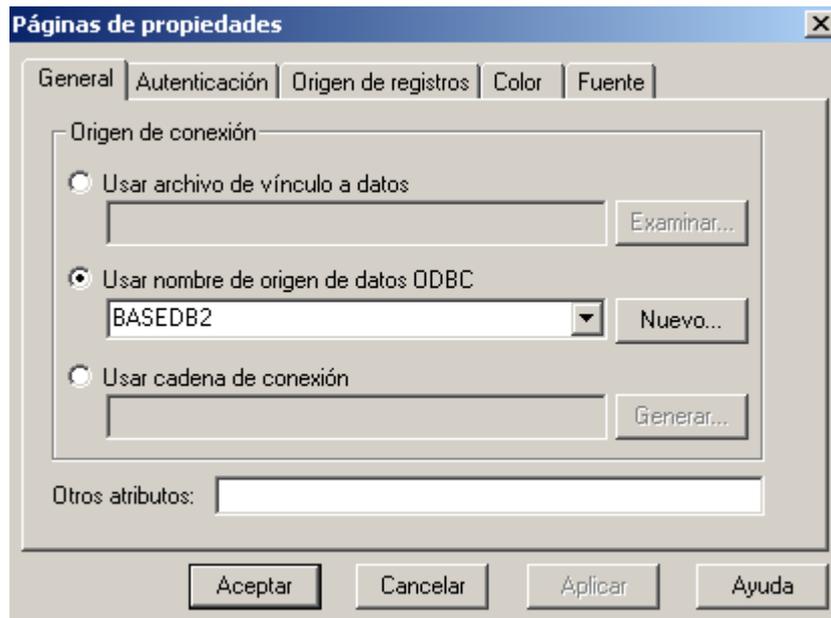
#### 2.5.1.4.1 Uso de las características en tiempo de diseño del Control **DataGrid**

Puede crear una aplicación para bases de datos con el control **DataGrid** sin necesidad de escribir una sola línea de código y aprovechar sus características en tiempo de diseño. Las siguientes instrucciones muestran un esquema general de los pasos necesarios para implementar el control **DataGrid** en un uso típico.

Para implementar un control **DataGrid** en tiempo de diseño

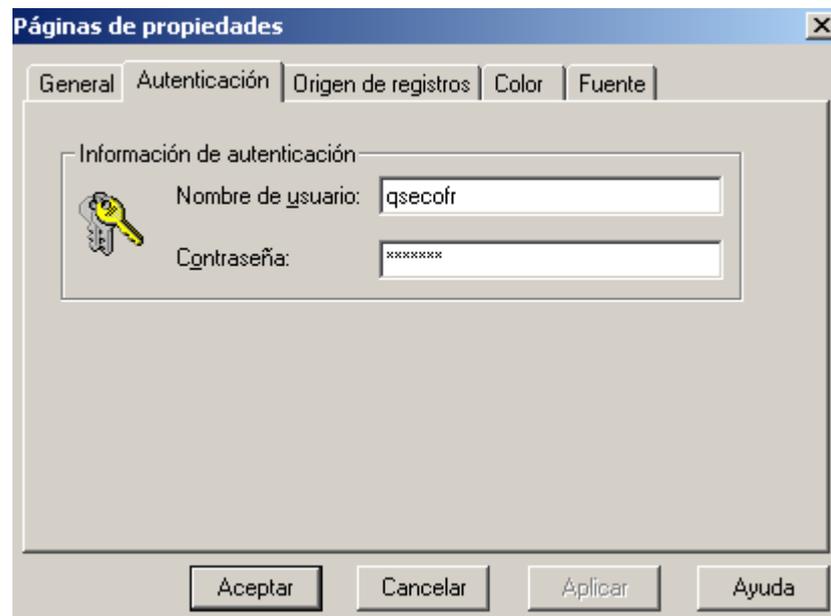
1. Sitúe un control de datos **ADO** en un formulario y establezca la propiedad **DataSourceName** al origen de datos ODBC creado en el paso 1.
2. En el campo **RecordSource** del control de datos **ADO**, escriba una instrucción SQL que devuelva un conjunto de registros:  

```
SELECT crpnom, crpnhc FROM crpacip WHERE crpnhc > 8000.
```
3. Sitúe un control **DataGrid** en un formulario y establezca la propiedad **DataSource** al **control de datos ADO**.
4. Haga clic con el botón secundario del *mouse* en **DataGrid** y después haga clic en **Propiedades**.
5. Mediante el cuadro de diálogo **Páginas de propiedades**, establezca las propiedades adecuadas del control para configurar la cuadrícula del modo en que quiera que aparezca y actúe. Así se obtendrá los datos, en este caso datos de DB2 al ejecutar el formulario con el **DBGrid** configurado. Las propiedades más importantes son:
  - Origen de conexión: se selecciona origen de datos ODBC para DB2.



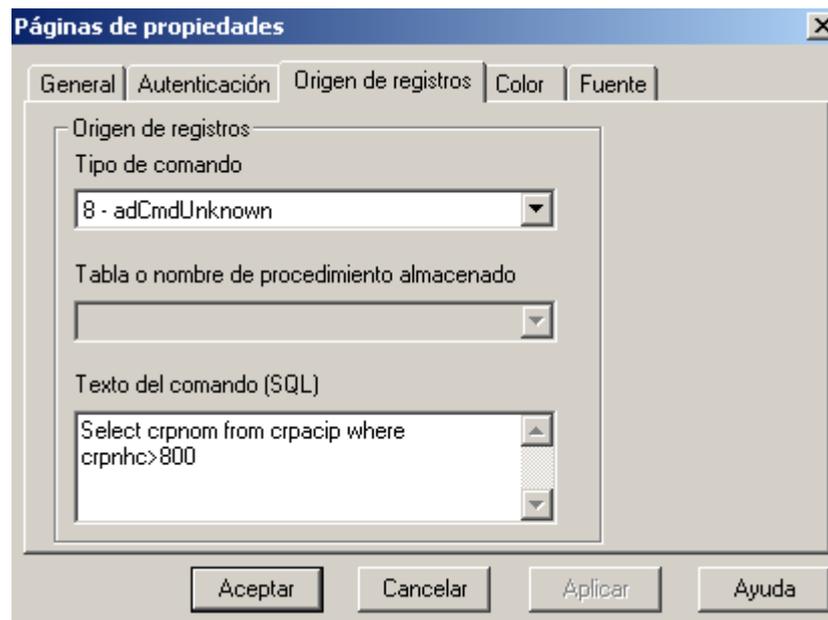
**Figura 2.18 Origen De Conexión.**

- Información de Autenticación: Nombre de usuario y contraseña de autenticación en AS/400.



**Figura 2.19 Autenticación.**

- Origen de registros: permite ingresar la sentencia de búsqueda SQL de resultados.



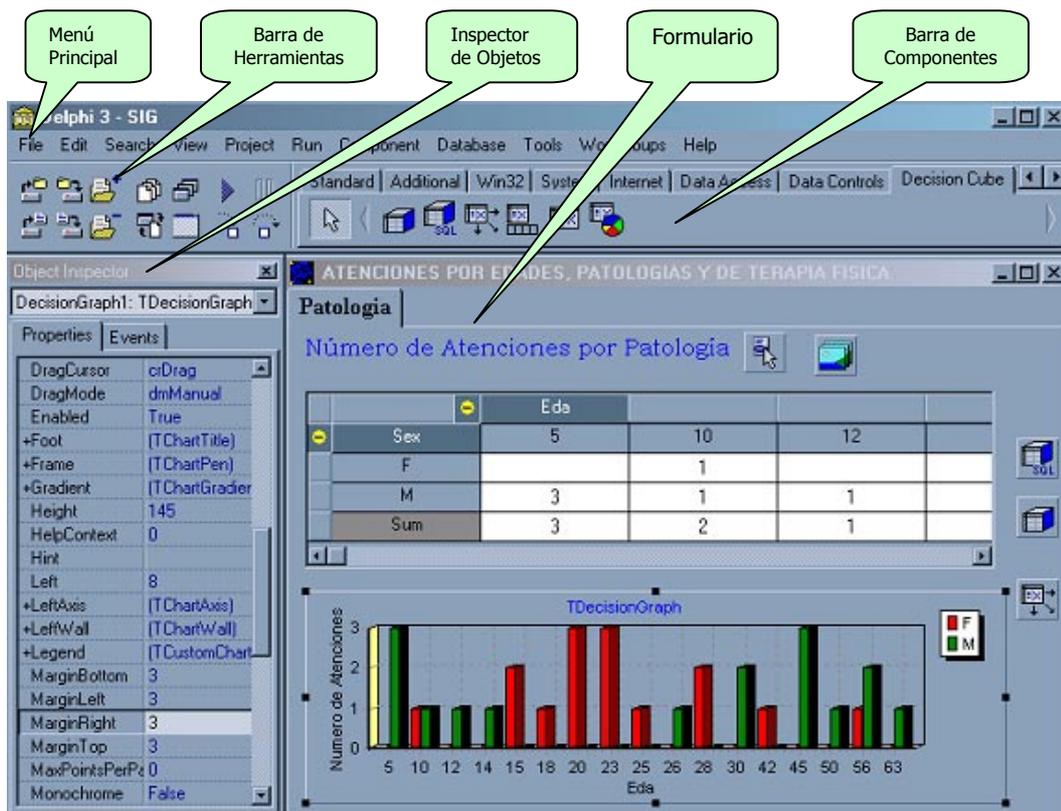
**Figura 2.20 Origen De Registros.**

Estos son los tres métodos que dispone Visual Basic para permitir la operación de datos DB2 en una aplicación de base de datos Windows.

**2.5.2 Acceso a datos de AS/400 desde Borland Delphi**

Delphi es una herramienta de desarrollo para aplicaciones Windows. La versión de análisis es la 3.0 cliente-servidor. Utiliza como lenguaje de programación Object Pascal. Dispone un entorno de desarrollo visual donde integra todas las herramientas necesarias para la creación de aplicaciones como editor de código, herramientas para conexiones a bases de datos, herramientas para Internet y todo tipo de controles estándares de Windows.

**2.5.2.1 Entorno De Desarrollo**



**Figura 2.21 Entorno De Desarrollo De Delphi En Ambiente De Diseño.**

**Menú Principal**

Se puede hacer prácticamente todo desde los diferentes submenús:

Agregar nuevas fichas al proyecto, reconfigurar el aspecto del entorno o definir las opciones de compilación de los ficheros ejecutables.

**Barra de Herramientas(SpeedBar)**

Es un conjunto de botones de acceso rápido a diferentes opciones de menú, lo que permite realizar las acciones mas frecuentes sin necesidad de navegar entre los menús.

Se puede añadir o eliminar elementos pulsando sobre la misma barra con el botón secundario del ratón.

### **Paleta de componentes**

La paleta de componentes da acceso a los componentes que conforman la biblioteca de componentes visuales, referida frecuentemente como VCL. Está dividida en diversas categorías según la función o tipo de componentes, de manera que se tiene en realidad un conjunto de 14 paletas o solapas, agrupadas en una misma interfase.

La inserción en el formulario de un componente seleccionado se puede realizar de dos maneras:

- Simplemente pulsando sobre el formulario y arrastrando el ratón para definir el tamaño.
- Pulsando dos veces en la propia barra de componentes.

### **Formulario (Form)**

La ficha, ventana, formulario, es la pieza básica del diseño de aplicaciones con Delphi. Podría describirse como el contenedor final sobre el que situaremos otros componentes; pero en realidad su funcionalidad va mucho más allá. Para empezar, es un componente más de Delphi, por lo que posee decenas de propiedades, métodos y eventos, mediante los cuales se puede modificar su aspecto: Título, color, tamaño, o configurar la funcionalidad de nuevas aplicaciones y la navegación entre ventanas.

### **Inspector de objetos (Object Inspector)**

El inspector de objetos permite modificar de manera sencilla las propiedades y respuestas a sucesos de los componentes. Sus datos muestran el estado del componente seleccionado o del mismo formulario. Está configurado como un cuadro de diálogo con dos páginas: propiedades y eventos.

La página de propiedades permite visualizar y modificar en tiempo de diseño las propiedades con las que cuenta un componente.

La página de eventos, enumera todos los eventos a los que el objeto seleccionado es capaz de responder.

## **Editor de código**

Delphi es una herramienta de desarrollo, por lo cual el editor de código es la parte principal del entorno, es aquí donde se escribirá el código del programa. Cada formulario de un proyecto tiene asociada una página de código.

El título de la ventana de edición muestra cual es la unidad de código visualizada. En la parte inferior de la ventana se muestra una barra de estado con tres paneles. En el primero se visualiza el número de línea y columna donde está situado el cursor de edición. En la segunda se indica si la unidad ha sido modificada o no. Finalmente en el tercer panel se indica el modo de escritura: inserción o sobre escritura.

### **2.5.2.2 La Biblioteca de componentes**

La biblioteca visual de componentes de Delphi (VCL) esta formada por objetos preconstruidos en Object Pascal que una vez ubicados en un formulario pueden ser modificados a la necesidad del usuario y así poder desarrollar variedad de aplicaciones Windows.

Además, Delphi permite a los desarrolladores construir sus propios componentes y añadirlos a la biblioteca de manera que aparecerá en la paleta de componentes, estando disponibles en cualquier proyecto a partir de ese momento.

Los componentes son objetos, con las características definidas en la programación orientada a objetos: encapsulamiento, abstracción, herencia y polimorfismo.

Cada componente de la biblioteca tiene sus características particulares; pero todos comparten características comunes con otros objetos de la jerarquía. En la cima de la arquitectura se encuentra el antecesor común a todos TComponent, que define las propiedades mínimas para que un componente pueda funcionar en el entorno de Delphi.

A continuación se detallan los componentes más importantes que se pueden utilizar para acceso a datos de AS/400 o de otras base de datos; clasificados en sus respectivas paletas:

#### **Paleta Estándar**

En esta paleta se encuentran los componentes mas utilizados y es la que aparece por defecto al iniciar Delphi. Ejemplo: Label, MainMenu, Edit, ListBox.

**Paleta Adicional**

En esta paleta se encuentran componentes que amplían a los estándar. Ejemplo: StringGrid, Image, Chart, etc.

**Paleta Win32**

En esta paleta se encuentran componentes que permitan crear aplicaciones que cumplan con las especificaciones de aspecto Windows 95 como por ejemplo: PageControl, ProgressBar, Animate, StatusBar, ToolBar, etc.

**Paleta Internet**

La paleta Internet que se encuentra disponible en la versión cliente-servidor de Delphi se encuentran componentes que gestionan los protocolos de acceso a la gran red. Ejemplo: ClientSocket, ServerSocket, NMFTP, SMTP, HTML, etc.

**Paleta Data Access**

En esta paleta data Access de Delphi se encuentran componentes para comunicarse con bases de datos locales y remotas y obtener la información que en ellas se almacena.

COMPONENTE	DESCRIPCIÓN	TIPO
DataSource	Realiza las conexiones a las bases de datos. Sirve para que diferentes controles que muestran los datos sean capaces de acceder a la base de datos mediante una interfaz común, independientemente de cómo se realice la conexión.	No visual(Los componentes no visuales tienen como función establecer una relación entre la aplicación y la base de datos no presentan datos).
Table	Permite la conexión a una tabla de base de datos a través de DataSource. Dispone de la propiedad DatabaseName que sirve para direccionar a la base de datos mediante su nombre.	No visual
Query	Permite crear y ejecutar consultas SQL a una base de datos a través de DataSource. También dispone de la propiedad DatabaseName	No visual
Database	Realiza conexiones a servidores remotos. Es uno de los controles que también posee la propiedad DatabaseName.	No visual

**Tabla 2.5 Componentes De Acceso A Datos De Delphi.**

## Paleta Data Controls

En esta paleta tenemos componentes que en general se utilizarán conectados a componentes Table o Query, albergando de esta forma los datos de las tablas y consultas. También permiten modificar los registros de las bases de datos.

COMPONENTE	DESCRIPCIÓN	TIPO
DBGrid	Presenta los datos en una rejilla de fila y columnas	Visual
BDNavigator	Posibilita la Navegación por los registros de una tabla o consulta, permitiendo su modificación.	Visual
DBText	Label vinculado a la base de datos	Visual
DBEdit	Cuadro de texto vinculado a la base de datos	Visual
DBListBox	Lista vinculada a la base de datos	Visual
DBChart	Objeto de representación con gráficos de datos de base de datos	Visual

**Tabla 2.6 Componentes De Presentación De Datos En Delphi.**

Los componentes mencionados pueden ser utilizados en el acceso a datos que se encuentran en DB2. Las operaciones que pueden hacerse son diversas, todas aquellas que se puedan realizar con SQL pero que no impliquen análisis multidimensional. Delphi si posee herramientas o componentes que permiten hacer el análisis multidimensional que no se ha presentado en esta sección, por esta razón que si se puede usar para desarrollar SIG.

Cada componente que disponga de la propiedad *DatabaseName* que hace referencia a la base de datos dispone ya de una lista desplegable de alias de bases de datos que permite Delphi y, además, los nombres de fuentes de datos ODBC que detecte que se encuentren en Windows. Esta es la forma como permite realizar el acceso a bases de datos.

A continuación se presenta un pequeño ejemplo de acceso a los datos usando los componentes mencionados anteriormente dentro de un formulario para acceder mediante una consulta SQL a tablas existentes en un sistema AS/400.

1. Desde luego lo primero es tener un origen de datos ODBC que exista en Windows y pertenezca a ODBC Client Access.
2. Se ubican dentro de un formulario de Delphi los siguientes controles: un DataSource y un Query de la paleta de *Data Access*, y un DBGrid de la paleta *Data Controls*.

3. Modificar la *propiedad DataSource* del objeto DBGrid del formulario con el nombre del DataSource que se dispone en el formulario que por defecto ya se encuentra en la lista desplegada al pulsar la propiedad del objeto.
4. Igualmente se modifica la propiedad *DataSet* del objeto DataSource con el nombre del Query que se ubica en el formulario. El nombre del Query se despliega en una lista al seleccionar esta propiedad DataSource del Query.
5. Se continúa con la modificación del objeto Query, seleccionando la propiedad *DatabaseName* del objeto ubicando el nombre del origen de datos AS/400 en la lista que se configuró anteriormente.
6. Presionando en la propiedad *SQL...* del objeto Query aparecerá el editor de código SQL. Se debe escribir la sentencia SQL adecuada para obtener los resultados deseados y pulsar Aceptar.
7. Se procederá a activar este objeto mediante su propiedad *Active = True*.
8. Si se ha logrado el acceso sin problemas en ese mismo instante se podrá visualizar los datos en el Grid del formulario sin necesidad de ejecutarlo.

Además de acceder a bases de datos utilizando los componentes visuales, también es posible hacerlo utilizando código escrito en Object Pascal en el que se manejan los métodos, eventos y propiedades de los componentes.



**Figura 2.22 Objetos De Acceso A Datos De Delphi En Tiempo De Diseño.**

### Paleta Decisión Cube

Contiene componentes pensados para facilitar el análisis de datos y simplificar la toma de decisiones.

COMPONENTE	DESCRIPCIÓN	TIPO
DecisionCube	Versión de <i>Table</i> que permite almacenar una estructura de datos multidimensional. Versión Cúbica de <i>Table</i> .	No visual
DecisionSource	Versión de <i>DataSource</i> para DecisionCube.	No visual
DecisionQuery	Permite crear y ejecutar consultas SQL a una base de datos a través de <i>DataSouce</i> . Versión cúbica de <i>Query</i> .	No visual
DecisionPivot	Presenta una interfaz que permite al usuario controlar los gráficos y rejillas de análisis.	Visual
DecisionGrid	Muestra el resultado del análisis en forma de rejilla	Visual
DecisionGraph	Muestra el resultado del análisis en forma de gráfico	Visual

**Tabla 2.7 Componentes Paleta DecisionCube.**

Los componentes de esta paleta son los componentes que se han usado en la elaboración del sistema de información gerencial que forma parte del aplicativo de la presente investigación.

Otras paletas que podemos mencionar son: QReport con componentes para diseño de reportes; paleta Dialogs para crear cuadros de diálogo; paleta ActiveX con controles OCX.