

CAPÍTULO VIII

VERIFICACIÓN DE LA HIPÓTESIS, CONCLUSIONES Y RECOMENDACIONES

8.1. Verificación de la Hipótesis

La hipótesis planteada al inicio de esta Tesis de Grado fue la siguiente:

“La creación de una metodología para manipular e implementar manejadores ODBC permitirá su rápida difusión, conocimiento y utilización para procesos de acceso y consulta de información en ambientes compartidos”.

Al finalizar el presente trabajo de tesis se ha podido desarrollar los siguientes resultados que permiten verificar la hipótesis planteada:

a. Se conoce específicamente la arquitectura ODBC.

Este material muestra un estudio completo de cuales son los componentes que forman parte de la arquitectura de un ODBC, como estos interactúan entre si, su funcionamiento y requerimientos para su utilización. En esta parte se da a conocer las gramáticas SQL que se utilizan, las funciones API ODBC y el manejo de los tipos de drivers ODBC para Base de Datos.

b. Descripción de los procesos que intervienen en el manejo de la comunicación entre bases de datos mediante ODBC.

Se desarrolló un material en donde se da a conocer en forma detallada el manejo de las opciones del Administrador de ODBC, el manejador ODBC específico utilizado en los prototipos, así como su configuración y el detalle de los parámetros que intervienen en una conexión ODBC. La forma como se crean los DSN tanto de usuario como de sistema, la instalación y configuración de los drivers ODBC de

diferentes fabricantes. Además el manejo del registro de Windows para dar a conocer los lugares y formas como se configuran las variables de registro que maneja un driver ODBC.

c. Manual o guía para el uso de manejadores ODBC que incluyen las bases de datos.

Este documento da a conocer una metodología para utilizar los manejadores ODBC que se instalan con las bases de datos.

Para ello se describen la instalación y configuración de los componentes necesarios para su comunicación tales como:

- Protocolos de comunicación TCP/P, IPX/SPX, NET8, WinSockets.
- El software de comunicación del servidor y el cliente.
- Los drivers ODBC.

Se describen también los objetos que permiten acceder a los datos remotos tales como: las conexiones, vistas remotas, etc.

d. Conocimiento de la programación de clases visuales (Visual C++) con las funciones de la API ODBC para un SDK particular.

Aquí se muestra la programación de manejadores ODBC con Visual C++. Esta programación es directamente en el nivel API ODBC, en donde se da a conocer el manejo de los tres elementos importantes de ODBC: el entorno, la conexión y la sentencia.

e. Manual o guía para la programación de manejadores ODBC con un lenguaje visual (C++) y un SDK.

Describe los componentes que se deben implementar para el diseño de un manejador, la interacción del mismo en una aplicación Cliente-Servidor. Tales como:

- Establecimiento de la conexión a la fuente de datos
- Configuración de los atributos del controlador
- Soporte para tipos de datos
- Preparación y ejecución de sentencias (basadas en la gramática soportada por el DBMS y/o en la necesaria para acceder a un almacén de datos particular no relacional)
- Obtención de la información del regreso de resultados
- Implementación de opciones avanzadas
- Terminación de transacciones
- Terminación de la conexión
- Tratamiento de los errores más comunes
- Instalación del controlador

f. Prototipo de aplicación para Administración Académica, mediante el uso de ORACLE, Visual FoxPro e interfaces ODBC.

Este prototipo muestra en forma práctica la aplicación de toda la metodología descrita en los puntos anteriores, hace uso de los drivers ODBC, los protocolos de comunicación TCP/IP y NET8 de Oracle, la configuración del servidor y el cliente, el uso de objetos de conexión de Visual FoxPro y acceso la base de datos Oracle mediante vistas remotas.

Este prototipo muestra el caso práctico del manejo de información que necesita una Institución Académica en su módulo de Tesorería, es decir en los procesos de matriculación de estudiantes. Todo el interfase de usuario está desarrollado en Visual FoxPro (Front End) y los datos se alojan en la BDD de ORACLE (Back End).

g. Prototipo de aplicación para Administración de Nomina de Personal mediante el uso de ASE Sybase, Visual Basic e interfaces ODBC.

El prototipo propuesto permite administrar un conjunto de empleados en lo que respecta a su currículum, puntaje de escalafón, vida laboral y nomina de pagos. La base de datos se encuentra en un servidor de BackEnd con Adaptive Server

Enterprise de Sybase sobre Windows NT 4.0, la comunicación de red es a través de TCP/IP y WinSockets, y el frontal está implementado con Microsoft Visual Basic 6.0 y ActiveX Data Control, la comunicación de datos se realiza a través de manejadores ODBC de Sybase e Intersolv. La metodología planteada funciona perfectamente para el presente aplicativo.

8.2. CONCLUSIONES

Al término de este trabajo de tesis se concluye lo siguiente:

El diseño de un controlador ODBC requiere conocimientos detallados de la arquitectura y manejo del motor de una Base de Datos o del almacén de datos en particular si no es un DBMS.

Se necesita conocer un lenguaje de programación de bajo nivel para acceder a almacenes de datos.

El uso de manejadores ODBC ha permitido hoy en día implementar Sistemas de Información con el uso de diferentes DBMS.

La programación de un manejador ODBC mediante el uso de lenguaje Visual C++ requiere de conocimientos avanzados del mismo.

El éxito de la comunicación con un almacén de datos depende de la forma como se realizó el diseño interno del driver ODBC.

Ya no es necesario comprar herramientas de interfaz de la misma marca que proporciona el motor de la base de datos con el uso de la tecnología ODBC.

Mediante el uso de tecnología ODBC se puede aprovechar la seguridad que proporcionan los DBMS y la diversidad de operaciones que brindan los lenguajes de programación.

El intercambio de información entre diferentes DBMS seguirá reforzándose a futuro con la mejoría en las técnicas de acceso, tipos de datos y estándares utilizados en los controladores ODBC.

Existen muchas herramientas CASE y aplicaciones que hacen usos de esta tecnología lo que confirma su difusión y utilización.

8.3. RECOMENDACIONES.

1. Utilizar este documento para orientarse en la utilización de manejadores ODBC existentes.
2. Aprovechar el uso de tecnología ODBC descrita a profundidad en esta tesis para crear herramientas mediante lenguajes de programación que realicen tareas especiales con los datos alojados en un determinado DBMS o en algún otra forma de almacenamiento, aprovechando la versatilidad y el bajo costo de producción de estas.
3. Usar esta tecnología para ofrecer nuevas alternativas de administración de información independiente de cualquier DBMS.
4. Se puede desarrollar utilitarios con esta tecnología que permitan realizar tareas específicas tales como:
 - Copiar información de una base de datos a otra.
 - Elaborar procesos de depuración de información.
 - Llevar determinada información al Web.
 - Acceder a datos en estructuras de almacenamiento no convencional.
 - Aprovechar información relegada por su inaccesibilidad.
 - Reducir costos al desarrollar manejadores propios.
 - Desarrollar manejadores especializados para cierto tipo de almacén de datos.
 - Mejorar los procesos de migración.

5. Mantener actualizadas las versiones de los drivers o controladores ODBC, para aprovechar las nuevas filosofías y estándares del manejo de datos.