



ADO.Net

Diseño y Programación Avanzada de Aplicaciones

Curso 2002-2003



índice

BORRADOR

Evolución del acceso a datos (II)

■ ODBC.

- Conectividad para Bases de Datos Abiertas
- Proveedor con más éxito hasta la fecha
- El grupo de acceso SQL (HP, DigitalSUn, Informix, IBM, Oracle y Microsoft) se crea para definir y promover estándares de acceso a datos.
- Objetivo Poder cambiar la BD sin cambiar el código
- En 1990 se define un API (Interfaz de aplicaciones) para acceso a datos

Evolución del acceso a datos (III)

■ ODBC (II).

- En 1992 Microsoft lanza ODBC 1.0.
- Proporciona una serie de funciones que permiten controlar el acceso a datos.
- Ofrece un conjunto de instrucciones para diferentes sistemas independiente de la plataforma, proveedor, BD y lenguaje que se emplee.
- No obstante el API ODBC es complejo y poco intuitivo

Evolución del acceso a datos (III)

■ DAO

- Es el primer modelo de datos en Visual Basic.
- Apareció con Visual Basic 3.0 (1992)
- Es una tecnología diseñada entorno al JET capaz de acceder a BD Access y base de datos ISAM(dBase,Paradox, FoxPro) y ODBC.
- Funcionaba muy bien accediendo a BD en el mismo PC, y no tanto en acceso a BD remotas

Evolución del acceso a datos (III)

■ RDO

- Apareció con Visual Basic 4.0 (1995)
- Objetivo: facilitar el acceso a BD remotas SQL-Server y Oracle.
- No tuvo éxito debido a la aparición de otras tecnologías mejores.

Evolución del acceso a datos (III)

■ ODBCDirect

- Apareció con Visual Basic 5.0 (1997)
- Alternativa al DAO
- Se encuentra entre el motor Jet y RDO
- Proporcionaba acceso directo a fuentes ODBC.
- Utiliza el motor de RDO pero con nombres DAO
- Ofrece mejor rendimiento en acceso a BD remotas, pero que deben ser ODBC

Evolución del acceso a datos (III)

■ OLE DB

- Apareció en 1996
- API de Microsoft para el acceso a datos tanto BD relacionales como no relacionales.
- Se basa en el modelo de componentes .COM
- Se divide en dos tipos de componentes:
consumidores y proveedores
 - Consumidores Utilizan los datos
 - Proveedores Hablan con las fuentes de datos y suministran los datos a los consumidores
- Solo accesibles desde C++.
- Modelo algo complejo

Evolución del acceso a datos (III)

■ ADO

- Objetos de datos ActiveX
- Aparece en 1996
- Es un envoltorio de OLE DB, que intenta ocultar su complicada sintaxis
- Cuenta con todas las prestaciones de OLE DB, o sea acceso a todo tipo de datos
- Se basa en una arquitectura con conexión, pero permite también trabajar sin conexión (datos en memoria)

Evolución del acceso a datos (III)

■ RDS

- Servicio de datos remota.
- Similar a ADO pero diseñado para proporcionar tecnología OLE DB para aplicaciones en la Web

BORRADOR

Evolución del acceso a datos (III)


■ ADO.Net


- Aparece en 2000
- Es la última tecnología ADO y RDS de microsoft
- Se construye en XML, el estándar de la industria
- Se puede controlar por cualquier aplicación que pueda leer dicho estándar independientemente de la plataforma y el lenguaje.

ADO.Net. Características

- Proporciona conectividad entre sistemas relacionales o no relacionales de un conjunto de componentes común.
- Permite el acceso a datos sin conexión
 - Tradicionalmente las aplicaciones Cliente/servidor durante su ejecución mantienen abierta una conexión con la BD.
 - Inconvenientes
 - Consumen valiosos recursos del sistema
 - Resulta muy complicado escalar aplicaciones.(Una aplicación de BD puede funcionar muy bien con 100 usuarios y muy mal con 1000).

- ADO.NET se definió para arquitecturas sin conexión.
- Pasos
 - Los datos se leen en un componente (*DataSet*)
- El Dataset **BORRADOR**
 - Almacena los datos aunque se rompa la conexión con la BD
 - La aplicación puede manipular los datos aunque no haya conexión
 - La BD se actualiza cuando se vuelve a tener conexión

- 
- La actualización se realiza de forma optimista.
 - Al reestablecer la conexión se comparan los datos originales, si alguien los ha cambiado genera un error y rechaza la operación.

- 
- ADO.net utiliza XML para transferir los datos.
 - El Dataset puede contener varias tablas(ADO con su recordset solo permitía una)
 - El dataset permite dar tanto una vista relacional como una vista jerárquica basada en XML.
 - Al utilizar datos en binario se podían bloquear los cortafuegos. XML es un formato basado en texto luego puede transferir datos a través de los cortafuegos de forma sencilla y fiable.

Principales componentes de ADO.net

