

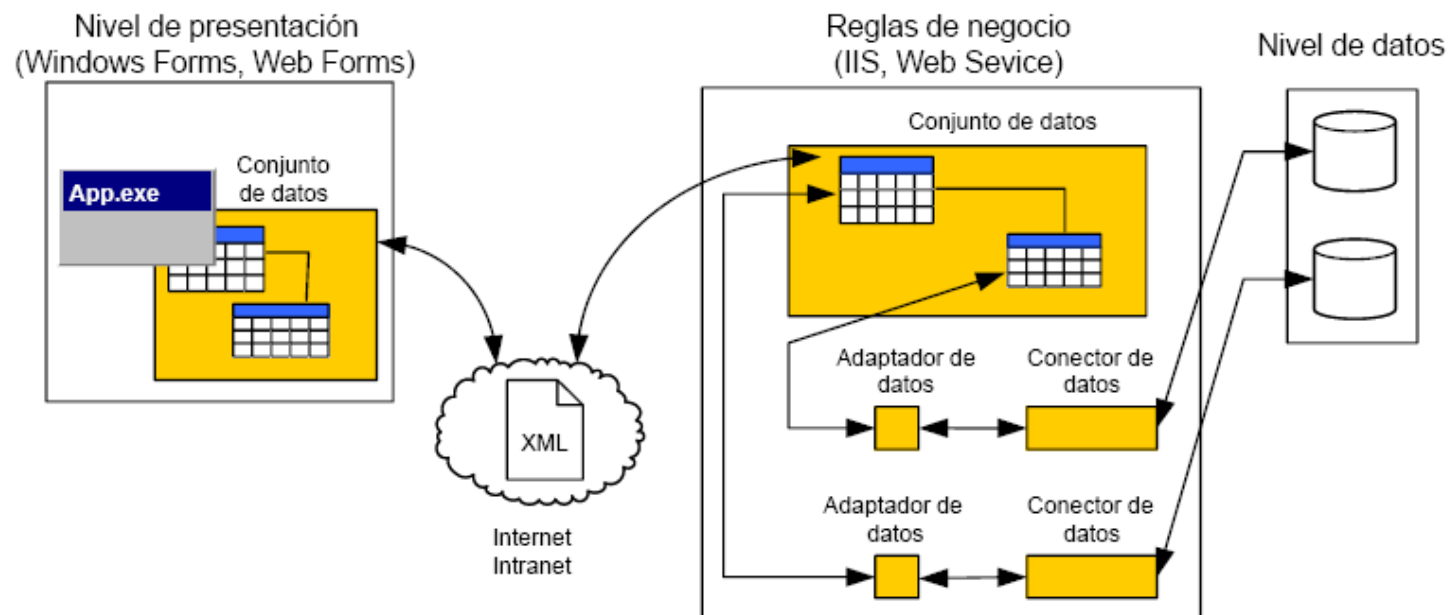
Tema 9: ADO.NET

- ☐ Introducción.
- ☐ Componentes ADO.NET.
- ☐ Proveedores de datos.
- ☐ Modelo de objetos.
- ☐ Espacio de nombres para ADO.NET.
- ☐ Modo conectado y modo desconectado.
- ☐ Conexión con la base de datos.
- ☐ Trabajar en modo conectado.
- ☐ Base de Datos MySql.

Introducción

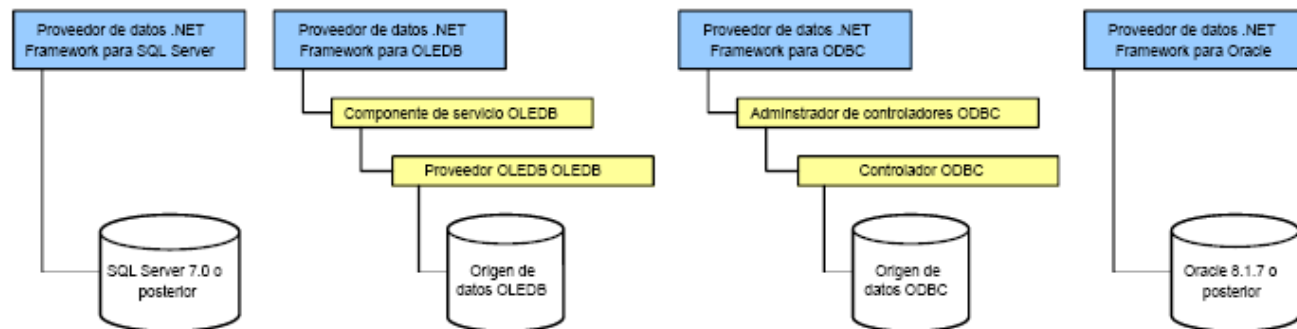
- ❑ ADO.NET es heredero de la tecnología ADO (ActiveX Data Objects) implantada por Microsoft hacia el año 2002.
- ❑ Supone la adaptación a .NET Framework de una arquitectura de base de datos que permitía acceder a bases de datos de cualquier proveedor ya se en modo local o remoto.
- ❑ ADO.NET.
 - Permite su utilización bajo cualquier entorno que soporte .NET Framework (no sólo bajo Windows).
 - Basada en las clases base de la arquitectura .NET.
 - Puede ser utilizada bajo cualquier lenguaje .NET.
 - Permite la ampliación a aplicaciones basadas en el modelo cliente/servidor.
 - Minimiza la carga de los servidores (modo desconectado).

Componentes ADO.NET



Proveedores de datos

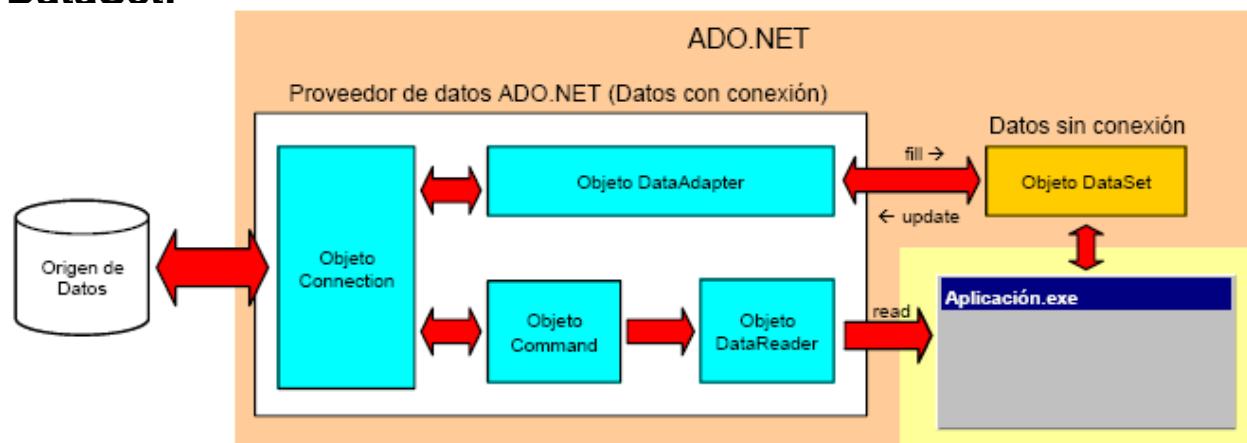
- ❑ Se utilizan para conectarse a la base de datos, recuperar información y ejecutar órdenes contra la misma.
- ❑ Dependientes del gestor de datos utilizado.
 - Proveedor de datos para SQL Server.
 - Proveedor de datos para OleDb.
 - Utiliza los controladores nativos OleDb para Windows.
 - Precisan de una capa adicional entre .NET y la base de datos.
 - Proveedor de datos para ODBC.
 - Proveedor de datos para Oracle (a partir de .NET Framework 1.1).



Modelo de Objetos

❑ Cinco objetos principales:

- En el proveedor de datos (dependen del gestor de bases de datos).
 - ✓ **Connection.**
 - ✓ **Command.**
 - ✓ **DataReader.**
 - ✓ **DataAdapter.**
- Independiente del proveedor de datos.
 - ✓ **DataSet.**



Modelo de Objetos (II)

☐ Objeto Connection.

- Establece la conexión con la base de datos mediante una “cadena de conexión”.

☐ Objeto Command.

- Ejecuta una acción contra el almacén de datos, ya sea de consulta o de acción.

☐ Objeto DataReader.

- Conjunto de registros recuperado a partir del objeto Command.

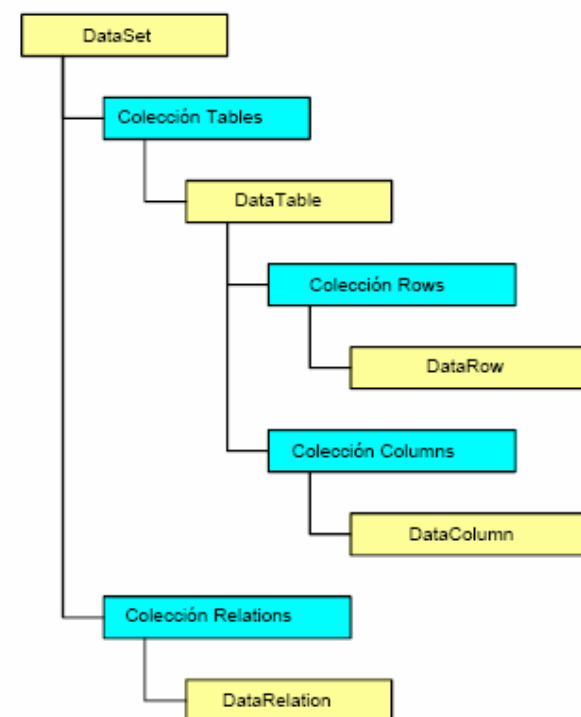
☐ Objeto DataAdapter.

- Puente entre la conexión y los datos almacenados en un DataSet.
- Permite cargar los datos en el DataSet a partir de un origen de datos y actualizarlos.

Modelo de Objetos (III)

❑ Objeto DataSet.

- Objeto “abstracto”: desligado de cualquier gestor de bases de datos.
- Conjunto de tablas obtenidas mediante el método Fill del objeto DataAdapter.
- Se puede considerar como una base de datos almacenada en la memoria caché del cliente.
 - ✓ Las tablas se cargan en la memoria caché del cliente, dejando disponible la conexión con el origen de datos para otros usuarios.



Espacio de nombres para ADO.NET

- ❑ Las clases de acceso a datos de .NET Framework están en el espacio de nombres System.Data.
 - Incluye las clases que no pertenecen a ningún proveedor específico.
- ❑ Los nombres de las clases pertenecientes al proveedor de datos .NET son nombres genéricos.
 - Existen clases distintas para cada sistema de gestión de bases de datos ubicadas en su propio espacio de nombres:
 - ✓ **System.Data.OleDb.** Contiene objetos asociados al proveedor de datos OleDb como OleDbConnection, OleDbCommand, OleDbDataReader y OleDbDataAdapter.
 - ✓ **System.Data.SqlClient.** Contiene objetos asociados al proveedor de datos SQL Server como SqlConnection, SqlCommand, SqlDataReader y SqlDataAdapter.
 - ✓ **System.Data.Odbc.** Contiene objetos asociados al proveedor de datos de ODBC como OdbcConnection, OdbcCommand, OdbcDataReader y OdbcDataAdapter.
 - ✓ **System.Data.OracleClient.** Contiene objetos asociados al proveedor de datos Oracle como OracleConnection, OracleCommand, OracleDataReader y OracleDataAdapter.

Modo conectado y modo desconectado

☐ Modo conectado.

- Utiliza los objetos Connection, Command y DataReader.
- Se establece una conexión permanente con el origen de datos.
- A partir de una conexión, el objeto Command generará un objeto DataReader con la información necesaria.
 - ✓ Los datos del objeto DataReader son de sólo lectura.
- El objeto Command, también se encargará de realizar las operaciones de actualización con la base de datos.
- La información entre el cliente y el servidor se establece en un formato binario propietario del gestor de base de datos.
- Se utiliza cuando se deben procesar pocos registros durante poco tiempo o cuando no sea necesario compartir la información con otras aplicaciones.
 - ✓ Realización de informes.
 - ✓ Páginas dinámicas ASP de Internet.

Modo conectado y modo desconectado (II)

- ❑ Modo desconectado.
 - La conexión sólo es necesario establecerla cuando se descarga información del origen de datos.
 - Cada tabla del DataSet precisa de un objeto DataAdapter.
 - ✓ El método Fill se encargará de cargar una tabla en el DataSet.
 - ✓ Los datos se almacenan en la memoria caché del cliente en un objeto DataSet.
 - ✓ Los datos almacenados en el DataSet se pueden modificar.
 - Las modificaciones efectuadas en el DataSet se pueden sincronizar con el origen de datos.
 - ✓ El método Update del objeto DataAdapter permite actualizar el origen de datos.
 - La información entre el cliente y el servidor se transmite en forma de datos XML (pueden ser utilizados por otra aplicación).
 - Se utiliza cuando:
 - ✓ Se necesita modificar los datos frecuentemente.
 - ✓ Es necesario que los datos estén mucho tiempo en memoria (por ejemplo en aplicaciones Windows Form).
 - ✓ Cuando no siempre es posible estar conectado al origen de datos (aplicaciones móviles).

Conexión con la base de datos

- ❑ Imprescindible tanto en modo conectado como desconectado.
- ❑ Pasos para crear una conexión:
 - Crear una instancia de algunas de las clases Connection.
 - Establecer la cadena de conexión mediante la propiedadConnectionString.
 - Abrir la conexión.
 - ✓ En modo conectado la conexión permanecerá abierta hasta que termine la aplicación.
 - ✓ En modo desconectado se cargarán los datos en un objeto DataSet y se cerrará la conexión.

Conexión con la base de datos (II)

☐ Crear una instancia de las clases.

- Para el proveedor de datos SQL.

Dim cnSQL As New SqlConnection

- Para el proveedor de datos OleDb.

Dim cnOleDb as New OleDbConnection

- Notas:

- ✓ Los nombres de las instancias serán cnSQL y cnOleDb respectivamente.
- ✓ Es necesario tener establecido el espacio de nombres o bien utilizar el nombre cualificado (System.Data.SqlClient.SqlConnection).
- ✓ Dependiendo del alcance que queramos dar a las variables utilizaremos los modificadores Dim, Private, Public, Friend, etc.

Conexión con la base de datos (III)

❑ Cadena de conexión.

- Todas las clases Connection de todos los proveedores tienen la propiedadConnectionString.

✓ El valor de la propiedad será una expresión de cadena formada por parejas de nombres de argumentos y valores, separados por un punto y coma.

Nombre_de_argumento= Valor

- Argumentos para una conexión para el proveedor de SQL Server.

Argumento	Valor
Data Source o Server	Nombre del servidor de base de datos.
Initial Catalog o Database	Nombre de la base de datos a la que se va a conectar
Integrated Security	Si se pone a false se debe especificar el nombre de usuario y la contraseña; si se pone a true se utilizarán las credenciales de la cuenta de Windows. Los valores permitidos son true, false, yes, no y spsi (recomendada, equivalente a true)
Persist Security Info	Si se pone a false (recomendado) la información de seguridad no se devuelve como parte de la conexión.
User ID	Nombre de usuario de una cuenta registrada en SQL Server
Pwd	Contraseña de inicio de sesión para una cuenta de SQL Server

Conexión con la base de datos (IV)

☐ Cadena de conexión (continuación).

- Argumentos para una conexión para el proveedor de OleDb para bases de datos Access.

Argumento	Valor
PROVIDER	Nombre del proveedor OleDb (para Access se utiliza Microsoft.Jet.OleDb.4.0)
Data Source	Especificación de archivo donde se encuentra el archivo .mdb

```
'Cadena de conexión para el proveedor SQLServer
cnSQL.ConnectionString = "Server=Servidor\SQLEXPRESS;Database=Ejemplo;" & _
                        "Integrated Security=sspi;Persist Security Info=false;" & _
                        "User ID=Salva;pwd="
```

```
'Cadena de conexión para el proveedor OLEDB para bases de datos Access
cnOleDB.ConnectionString = "PROVIDER=Microsoft.Jet.OleDb.4.0; " & _
                          "Data Source=c:\BBDD\ejemplo.mdb"
```

☐ Abrir y cerrar la conexión.

- cnOleDb.Open()
- cnOleDb.Close()

Trabajar en modo conectado

❑ Recuperar filas.

- Se realiza con el método **ExecuteReader** de la clase Command.
 - ✓ **Command** es una clase del proveedor de datos, por lo que es preciso elegir la subclase adecuada (OleDbCommand, SqlCommand, OdbcCommand u OracleCommand).
 - ✓ En el constructor se pasaría la sentencia SQL de recuperación (sentencia SELECT) necesaria para recuperar filas.
 - ✓ ExecuteReader devuelve un objeto de la clase **DataReader**.
 - También es una clase del proveedor de datos.
 - Representa el conjunto de filas recuperados por la SELECT.
 - Es sólo de lectura y sólo se puede ir a la siguiente fila.

Trabajar en modo conectado (II)

- ❑ Ejemplo de obtención de un DataReader a través de la ejecución del método ExecuteReader de la clase Command.

```
'Cadena de conexión para el proveedor SQLServer
Dim cnSQL As New SqlConnection
cnSQL.ConnectionString = "Server=Servidor\SQLEXPRESS;Database=Ejemplo;" & _
                        "Integrated Security=sspi;Persist Security Info=yes;User ID=Salva;pwd="
cnSQL.Open()

'Crear una orden de recuperación
Dim miOrden As SqlCommand = New SqlCommand("SELECT * FROM Clientes", cnSQL)

'Crear el DataReader
Dim drClientes As SqlDataReader

'Ejecutar la consulta
drClientes = miOrden.ExecuteReader
```

- ✓ En este momento disponemos de toda la información en el DataReader

Trabajar en modo conectado (III)

☐ Acceder a las filas recuperadas.

- El método Read carga en el objeto DataReader la siguiente fila.
 - ✓ Devuelve false si no hay más filas.

- Accedemos al contenidos de las columnas mediante el nombre o la posicion.

```
While drClientes.Read()  
    Listbox1.items.add(drClientes("nombre"))  
End While  
drClientes.Close()
```

- Vamos moviendonos por el DataReader con el método Read hasta llegar al último registro.

Trabajar en modo conectado (IV)

☐ Ejecutar funciones agregadas.

- El método **ExecuteScalar** de la clase Command devuelve sólo la primera columna de una sentencia SELECT.

```
'Obtener el número total de registros de la tabla dbo.Clientes  
miOrden = New SqlCommand("SELECT COUNT(*) FROM Clientes", cnSQL)  
Msgbox ("Número de filas: " & miOrden.ExecuteScalar)
```

☐ Ejecutar ordenes SQL de actualización.

- Se utiliza el método **ExecuteNonQuery** de la clase Command.
- Se la sentencia SQL con la que se cree el objeto Command.

Trabajar en modo conectado (V)

- ❑ Ejemplo de ordenes SQL de actualización: update, insert y delete a través del método ExecuteNonQuery de la clase Command.

'Modificar la Ciudad del cliente con código 10101

```
miOrden = New SqlCommand("UPDATE Clientes SET Ciudad='Altea' WHERE IdCliente=10101", cnSQL)
miOrden.ExecuteNonQuery()
```

'Añadir el cliente con IdCliente 10104

```
miOrden = New SqlCommand("INSERT INTO Clientes " & _
                        "(IdCliente,Apellido,Nombre,Ciudad,Provincia)" & _
                        "VALUES (10104,'Ordóñez','Manuel','Tarrasa','Barcelona','08227')", cnSQL)
miOrden.ExecuteNonQuery()
```

'Eliminar el cliente con código 10298

```
miOrden = New SqlCommand("DELETE FROM Clientes WHERE IdCliente=10298", cnSQL)
miOrden.ExecuteNonQuery()
```

Base de Datos MySql

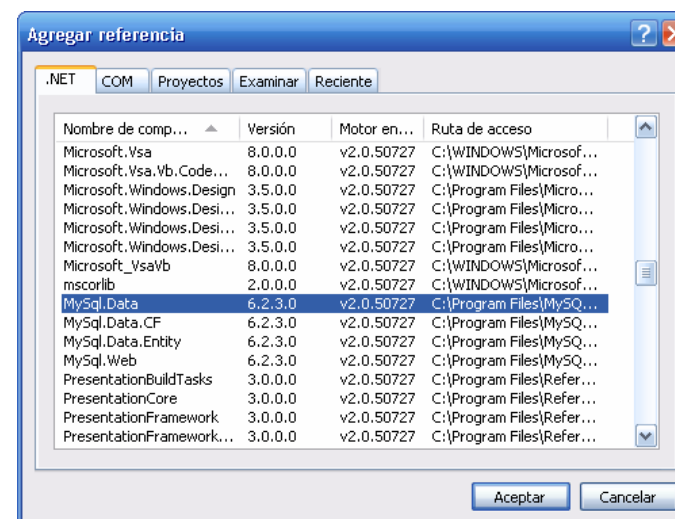
❑ Mysql.

- Descargar de la página de MySql (Oracle) [mysql-connector-net-x.x.x.zip](#)
- Agregar referencia MySql.Data
- Espacio de nombres para MySql

Imports MySql.Data.MySqlClient

- Disponible los objetos necesarios para trabajar con MySql

- ✓ Conexión
- ✓ Comando
- ✓ DataReader
- ✓ DataAdapter



Base de Datos MySql (II)

☐ Objeto Connection

Dim cn As New **MySqlConnection**

cn.ConnectionString = "server=localhost; user id=root; password=pwd; database=nombre_bd"

cn.ConnectionString = "server=localhost; user id=root; password=""; database=master"

☐ Objeto Command (**ExecuteReader**, **ExecuteScalar** y **ExecuteNonQuery**)

Dim miorden As **MySqlCommand** = New MySqlCommand(SQL, conexion)

Dim miorden As MySqlCommand = New MySqlCommand("select * from usuarios", cn)

☐ Objeto DataReader

Dim drusuarios As **MySqlDataReader**

☐ Objeto DataAdapter

Dim drusuarios As **MySqlDataAdapter**