

ADO .Net.

ADO.NET es un conjunto de clases para trabajar con datos. Cuando trabajamos con datos, utilizamos varios objetos para recuperar y modificar información de una base de datos. En general, cuando trabajamos con bases de datos con ADO.NET, podemos contemplar dos escenarios: acceso conectado y acceso desconectado.

Acceso conectado y desconectado.

Un **entorno conectado** es aquel en el que un usuario o una aplicación están conectados continuamente a una fuente de datos.

Proporciona las siguientes ventajas:

- Un entorno conectado es más fácil de mantener.
- La concurrencia se controla más fácilmente.
- Es más probable que los datos estén más actualizados que en un escenario desconectado.

Y los siguientes inconvenientes:

- Debe mantenerse una conexión de red constante.
- Un escenario conectado proporciona una escalabilidad limitada.

Un ejemplo en el que debe utilizarse una conexión continua: procesos que requieren una conexión en tiempo real, donde los datos deben estar constantemente actualizados.

Un **entorno desconectado** es aquel en el que un usuario o una aplicación no están conectados constantemente a una fuente de datos, se abre la conexión, se recuperan los datos y la conexión se cierra. El usuario trabaja con los datos y la conexión vuelve a abrirse para actualizar otras peticiones, tenemos las siguientes ventajas:

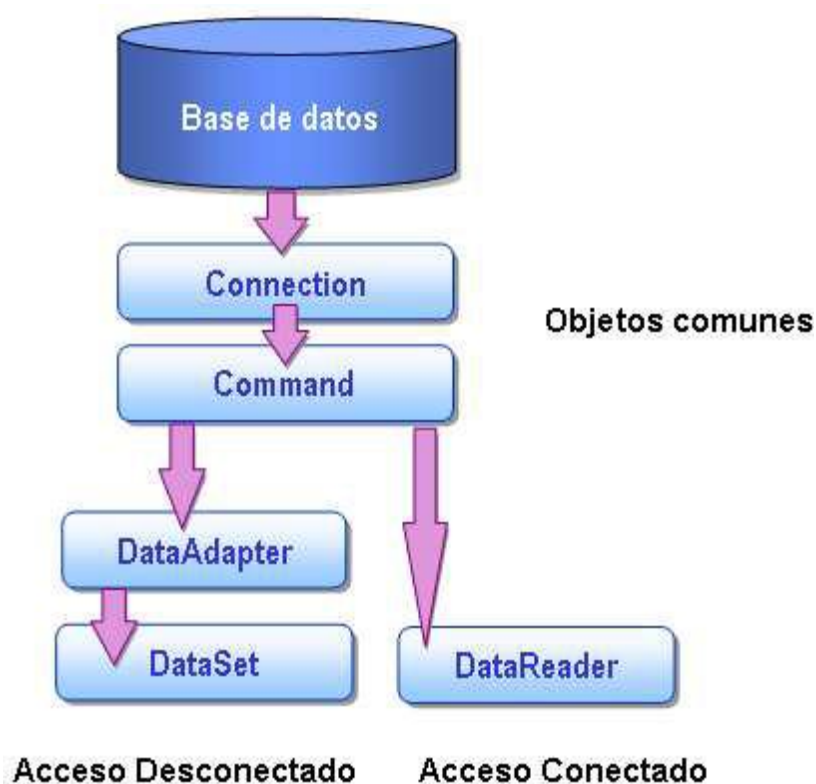
- Las conexiones se utilizan durante el menor tiempo posible, permitiendo que menos conexiones den servicio a más usuarios.
- Un entorno desconectado mejora la escalabilidad y el rendimiento de las aplicaciones, maximizando la disponibilidad de conexiones.

Como inconvenientes:

- Los datos no siempre están actualizados.
- Pueden producirse conflictos de cambios que deben solucionarse.
- La transmisión de datos puede percibirse más lenta de lo que sería en entornos conectados.

En un entorno desconectado, varios usuarios pueden modificar los datos de los mismos registros al mismo tiempo; por ello, nuestra aplicación debe gestionar conflictos en las actualizaciones de datos.

Clases ADO .Net.



System.Data y proveedores de acceso a datos.

Además del método de conexión en ADO .Net cabe distinguir el espacio de nombres donde reside cada objeto de los arriba mencionados:

Las clases del namespace **System.Data** son bastante extensas y variadas la más importante es la clase **DataSet**. Las clases de este espacio de nombres son comunes para cualquier bases de datos es decir utilizaremos de igual forma para **Access**, **Mysql**, **SqlServer**, etc.

Los **proveedores de acceso a datos** es la capa inferior de la parte correspondiente al acceso de datos y es la responsable de establecer la comunicación con las fuentes de datos. En este conjunto de namespaces, encontraremos las clases Connection, Command, DataAdapter y DataReader. En la instalación de Visual Studio se encuentran por defecto los siguientes:

- ODBC .NET Data Provider
- OLE DB .NET Data Provider
- Oracle Client .NET Data Provider
- SQL Server .NET Data Provider

Estos proveedores de acceso a datos incluidos, los podemos encontrar en los namespaces:

- **System.Data.Odbc:** permite conectar nuestras aplicaciones a fuentes de datos a través de ODBC, **ya en desuso**.
- **System.Data.OleDb:** permite conectar nuestras aplicaciones a fuentes de datos a través de OLE DB como **Access**.
- **System.Data.OracleClient:** es un proveedor de acceso a datos especialmente diseñado para bases de datos **Oracle**.
- **System.Data.SqlClient:** es un proveedor de acceso a datos nativo, que nos permite conectar nuestras aplicaciones a fuentes de datos **Microsoft SQL Server**

Sin embargo, hay muchos motores de bases de datos de igual importancia como **MySQL**, AS/400, etc. En estos casos, si queremos utilizar un proveedor de acceso a datos nativo, deberemos acudir al fabricante o iniciativas particulares para que nos proporcionen el conjunto de clases necesarias que nos permitan abordar esta acción. En este curso instalaremos el proveedor de MySQL.

De todas formas, usemos el proveedor de acceso a datos que usemos, la forma de trabajar con ellos es la misma, todas las clases tiene los mismos métodos y propiedades y hasta se llaman igual, por ejemplo el objeto **connection** para Access se llama OleDbConnection y el de **MySQL** se llama **MySQLConnection**.

Métodos de trabajo.

Independientemente de la base de datos (menos en el caso de MySQL) podemos trabajar con las clases antes descritas de dos formas: usando asistentes y herramientas visuales o usando código.

En el primer caso usaremos la ventana del *explorador de bases de datos* y la ventana de *orígenes de datos*, que nos permitirá mostrar y modificar tablas sin escribir código, desafortunadamente **no siempre podremos realizar todas las tareas usando estas herramientas y tendremos que recurrir a instanciar las clases y trabajar con ellas con código.**

En este módulo vamos a aprender conectarnos a MySQL y SqlServer.