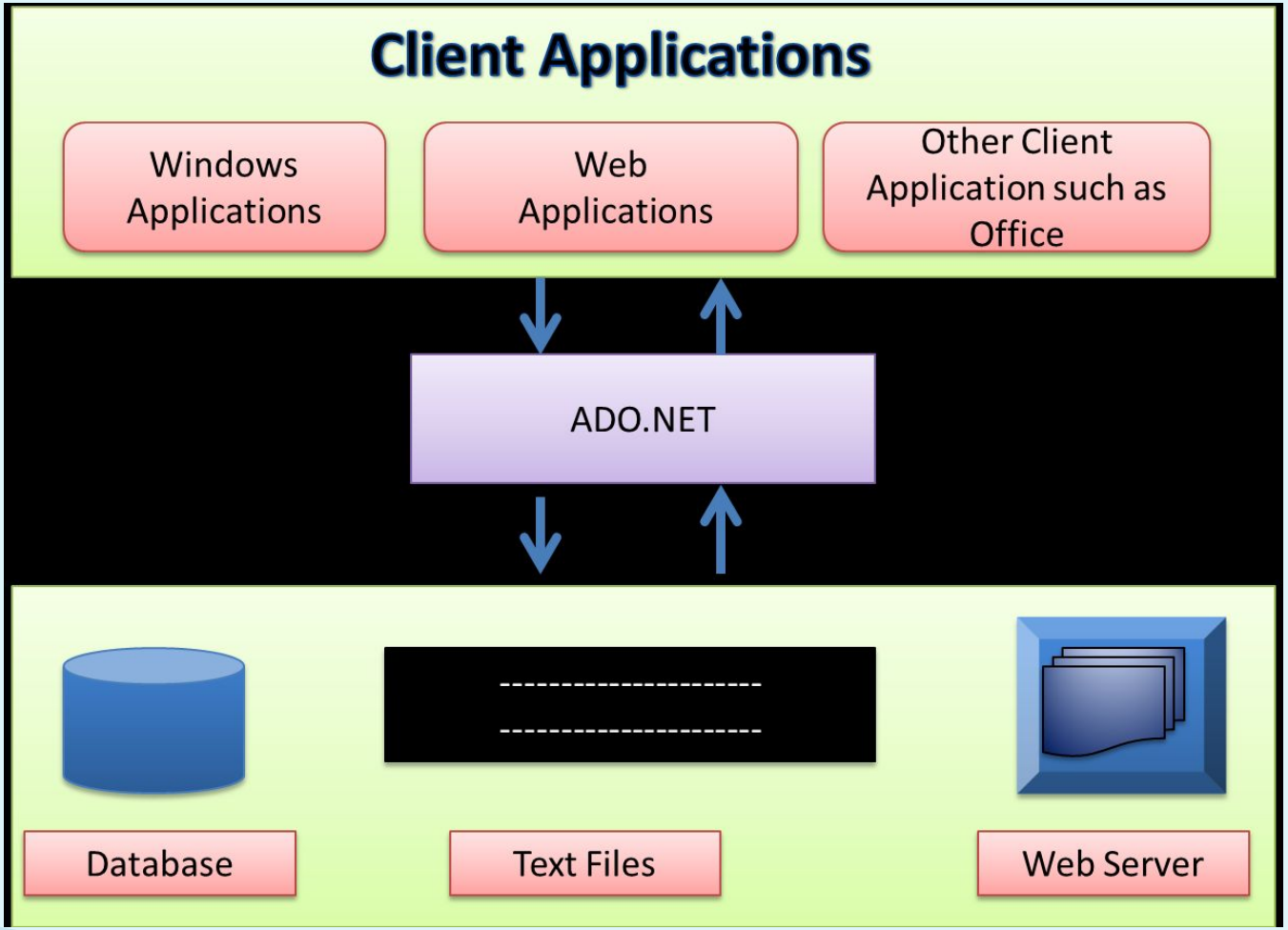
ADO.NET

¿QUÉ ES ADO.NET?

- ActiveX Data Objects: e**s una parte del framework de .NET que se encarga del acceso a los datos (BBDD) de nuestra aplicación.**
- **La BBDD puede ser SQLServer, MySql, Oracle, XML, etc.**
- **Se puede usar con WindowsForms, WPF, ASP.NET, ASP.NET Core, UWP, etc.**
- **UWP puede usar ADO.Net desde la Fall Creator Update.**
- **Antes de esa actualización UWP usaba SQLite o acceso a una API REST mediante Json.**

¿QUÉ ES ADO.NET?



ADO.NET Y ENTITY FRAMEWORK

- Podemos trabajar con ADO.NET de dos formas diferentes:
  - La primera es usando ADO.NET de la manera “tradicional”. Esto es implementando nosotros mismos por código la conexión con la BBDD, las llamadas a los procedimientos, las instrucciones SQL, etc.
  - La segunda forma es mediante el uso del ORM Entity Framework. Mediante este ORM, podremos conectarnos a una BBDD y realizar las operaciones más comunes de una manera más sencilla.

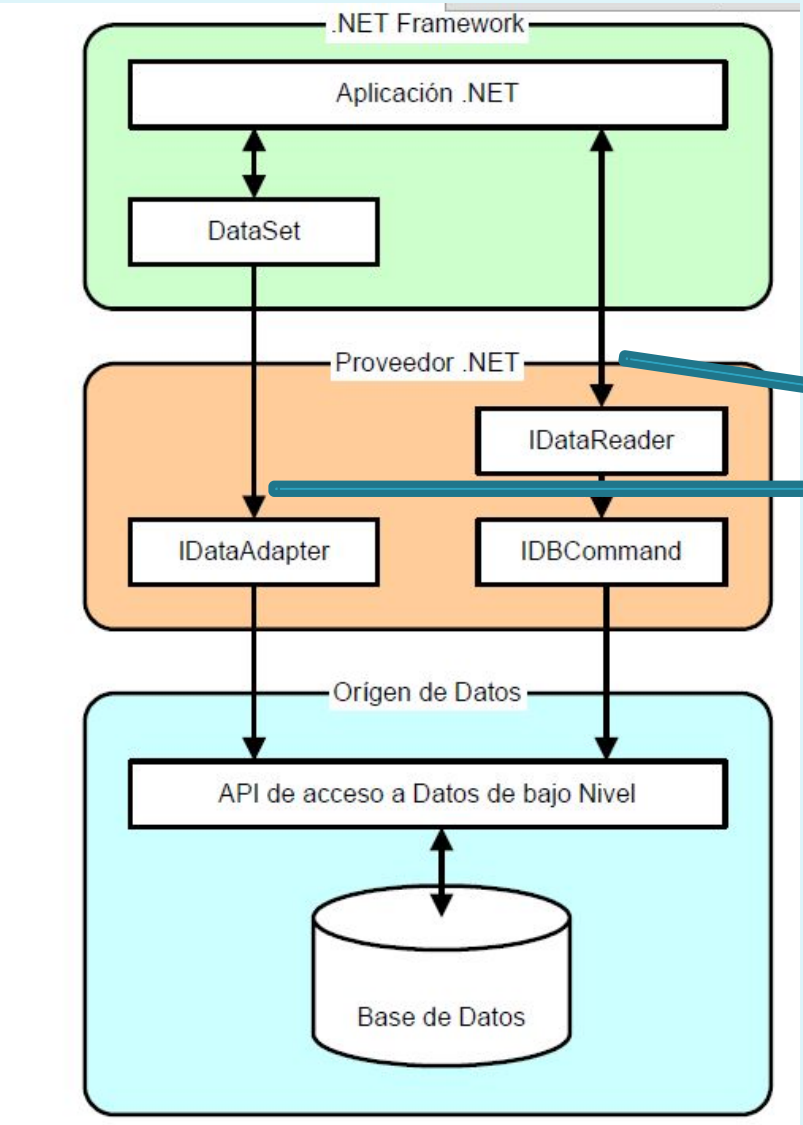
    Este es un enlace a un ejemplo de cómo se crean las entidades a partir de la BBDD en Entity Framework

https://www.entityframeworktutorial.net/entityframework6/create-entity-data-model.aspx

Y este es un enlace con un ejemplo de un CRUD

https://www.geeksforgeeks.org/basic-crud-create-read-update-delete-in-asp-net-mvc-using-c-sharp-and-entity-framework/

ADO.NET CLÁSICO



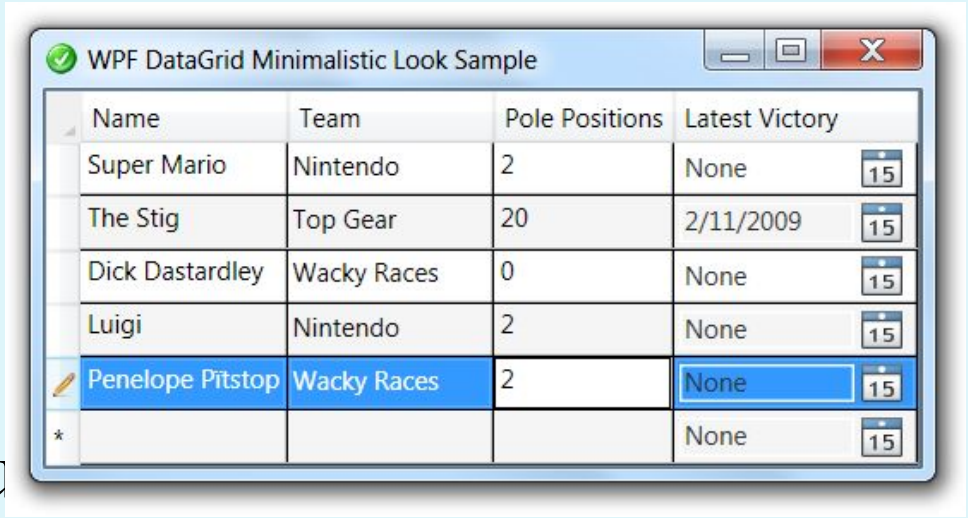
- Cuando trabajamos en ADO.NET clásico, podemos hacerlo de dos formas diferentes:
  - Modo Conectado
  - Modo desconectado

MODO DESCONECTADO

- Mediante un DataAdapter, nos conectaremos a una BBDD y realizaremos una consulta SQL para “traernos” un conjunto de datos.
- Ese conjunto de datos se guardan en la memoria mediante un objeto DataSet.
- Un DataSet es un conjunto de tablas, relaciones, etc. Es como una pequeña parte de la BBDD en nuestra memoria.
- Realizaremos cambios en nuestra aplicación sobre esos datos (borrado, actualización, etc.).
- Finalmente, mandaremos actualizar la BBDD con los cambios que se hayan producido en el DataSet.
- Nosotros no realizaremos las sentencias SQL, nos apoyaremos en un CommandBuilder.

DataGrid

- La forma más común de mostrar y editar los datos de un DataSet es mediante un control DataGrid.



- Este control se suele u
WPF.
- El control DataGrid está disponible en UWP desde 2018.

¿QUÉ USAREMOS EN ASP.NET?

- ADO.NET en modo desconectado es apropiado para aplicaciones en las que contemos con un ordenador y su memoria. Windows Forms, WPF, UWP.
- Esto no pasa en aplicaciones Web, donde la memoria que usarían TODOS los objetos DataSets de TODOS los usuarios conectados a nuestra aplicación, sería la memoria de nuestro servidor.
- Esto hace que lo mejor sea usar ADO.Net en modo conectado en aplicaciones ASP.Net.

ADO.Net MODO CONECTADO

- No tiene el objeto DataSet. Tenemos uno menos potente pero menos costoso: DataReader.
- Mediante un DataReader nos “traeremos” los datos de una fuente de datos.
- Mediante el objeto Connection estableceremos la conexión.
- Mediante el objeto Command, realizaremos sentencias de tipo Insert, Update o Delete.

Connection

- Establece una sesión con una fuente de datos.
- Implementada por **SqlDbConnection,ODBCConnection**, **OleDbConnection**, etc.
- Funcionalidad
  - Abrir y Cerrar conexiones
  - Gestionar Transacciones
- Usada conjuntamente con objetos **DbCommand** y **DataAdapter.**

ABRIR UNA CONEXIÓN

```
using Microsoft.Data.SqlClient;
…..
SqlConnection miConexion= new SqlConnection();

 try
     {
miConexion.ConnectionString
="server=localhost;database=nombreBBDD;uid=prueba;pwd=123;“;
miConexion.Open();
}
Catch….
```

EJERCICIO

- Realizar el ejercicio 1 de la unidad.

HACER UNA CONSULTA

```
SqlConnection miConexion= new SqlConnection();
List<clsPersona> listadoPersonas= new List<clsPersona>();
SqlCommand miComando = new SqlCommand();
SqlDataReader miLector;
clsPersona oPersona;
miConexion.ConnectionString
=("server=localhost;database=nombreBBDD;uid=prueba;pwd=123;“
        try
        {
    miConexion.Open();
          //Creamos el comando (Creamos el comando, le pasamos la sentencia y la conexion, y
lo ejecutamos)
          miComando.CommandText = "SELECT * FROM personas";
          miComando.Connection = miConexion;
          miLector = miComando.ExecuteReader();
……..……
```

HACER UNA CONSULTA

```
……….
miLector = miComando.ExecuteReader();
            //Si hay lineas en el lector
            if (miLector.HasRows)
            {
               while (miLector.Read())
               {
                  oPersona = new clsPersona();
                  oPersona.id = (int)miLector["IDPersona"];
                  oPersona.nombre = (string)miLector["nombre"];
                  oPersona.apellidos = (string)miLector["apellidos"]
//Si sospechamos que el campo puede ser Null en la BBDD
                     if (miLector["fechaNac"] != System.DBNull.Value)
                       {oPersona.fechaNac = (DateTime)miLector["fechaNac"];}
                  oPersona.direccion = (string)miLector["direccion"];
                  oPersona.telefono = (string)miLector["telefono"];
                  listadoPersonas.Add(oPersona);
               }
            }
            miLector.Close();
            miConexion.Close();
         }
         catch (SqlException exSql)
         {
            throw exSql;
}
```

EJERCICIO

- Realizar el ejercicio 2 de la unidad.

REALIZAR UN COMANDO (BORRAR)

```
public int deletePersonaDAL(int id)
        {
                int numeroFilasAfectadas= 0;
        SqlConnection miConexion= new SqlConnection();
        SqlCommand miComando = new SqlCommand();
        miConexion.ConnectionString =("server=localhost;database=nombreBBDD;uid=prueba;pwd=123;“
        miComando.Parameters.Add("@id", System.Data.SqlDbType.Int).Value = id;
          try
          {
         miConexion.Open();
            miComando.CommandText = "DELETE FROM Personas WHERE IDPersona=@id";
            miComando.Connection = miConexion;
            numeroFilasAfectadas= miComando.ExecuteNonQuery();
          }
          catch (Exception ex)
          {
            throw ex;
          }
          return numeroFilasAfectadas;
        }
```

EJERCICIO

- Realizar el ejercicio 3 de la unidad