Creamos la clase “PokemonNegocio”, donde se crean los ***métodos de acceso a datos para los pokemons.***

Si yo tuviese otra entidad, como entidad Persona, entidad Auto, entidad Elemento y necesito hacer un par de métodos de acceso a datos, cada clase va tener la clase de Negocio con sus métodos de acceso a datos.

Los métodos tienen que ser “public” para que tenga acceso desde el exterior

Para establecer la conexión y leer los datos necesito declarar ciertos objetos y configurarlos para poder establecer la conexión. Para esto debo incluir una librería

Using System.Data.SqlClient;

A partir del momento que declaro la utilización de esta librería voy a poder crear los siguientes objetos:

* SqlConnection: para conectarme a algún lado

SqlConnection conexion = new SqlConnection();

*Creo un objeto del tipo SqlConnection llamado conexión*

* SqlCommand: para realizar acciones

SqlCommand comando = new SqlCommand();

* SqlDataReader: Como resultado de la lectura a la DB, voy a obtener un set de datos, y eso lo alojo en un lector

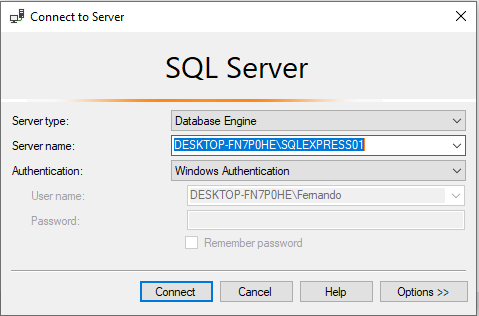
SqlDataReader lector;

No le generamos una instancia, porque cuando realice la lectura voy a obtener como resultado una instancia de un objeto del tipo SqlDataReader

Luego tenemos que configurar en el “try/catch”.

1. Lo primero es la cadena de conexión con el motor de la DB local.

conexion.ConnectionString = "server=DESKTOP-FN7P0HE\\SQLEXPRESS01";



Para uso general usamos la siguiente dirección: "server=.\\SQLEXPRESS01"

Esto es así para no tener que estar modificando el nombre de la máquina

Luego configuramos a que base de datos:

conexion.ConnectionString = "server=DESKTOP-FN7P0HE\\SQLEXPRESS01; database=POKEDEX\_DB";

Ahora configuramos como me voy a conectar. Hay dos maneras, una por seguridad integrada (la propia de Windows), la otra es por SQL Server Authentication, pero para esta ultima necesitas tener un usuario y contraseña del motor de la BD. Lo nuestro quedaría con la opción 1:

conexion.ConnectionString = "server=DESKTOP-FN7P0HE\\SQLEXPRESS01; database=POKEDEX\_DB; integrated security=true";

Si tuviésemos Usuario y Contraseña, se pone FALSE; usuario: \*\*\*\* password:\*\*\*

1. Ahora configuramos el comando que sirve para realizar la acción, en este caso vamos a realizar la “lectura”. Esta la voy hacer mandando la sentencia SQL que yo quiero ejecutar.

Indicamos el tipo de comando que yo quiero indicar con CommandType. Hay 3 tipos:

* Tipo texto
* Tipo procedimiento almacenado: para que ejecute una f(x) guardada en BD
* Tipo enlace directo con la tabla

comando.CommandType=System.Data.CommandType.Text;

Luego tenemos que indicar cual va ser el texto, que va ser la consulta SQL

comando.CommandText = "Select Numero, Nombre, Descripcion From POKEMONS";

Lo siguiente es decirle al comando.Connection que va ejecutar ese comando es esta “conexión”

comando.Connection = conexion;

Luego abro la conexión:

conexion.Open();

Realizo la lectura y esto me da como resultado un tipo de dato SqlDataReader, por lo que lo guardo en mi variable “lector”:

lector=comando.ExecuteReader();

Luego hago un bucle para que me vaya leyendo fila por fila

while(lector.Read())

{

Pokemon aux = new Pokemon();

aux.Numero = lector.GetInt32(0);

aux.Nombre = (string)lector["Nombre"];

aux.Descripcion = (string)lector["Descripcion"];

lista.Add(aux);

}

En cada vuelta, cada vez que haga “lector.Read()” y de TRUE (o sea encuentre dato) va bajar al siguiente. En cada vuelta va crear un nuevo objeto (aux) donde le va a guardar los datos de la línea que corresponda y al final la guarda en la “lista”. Cuando no haya mas el “lector.Read” va dar FALSE y va a retornar la lista.

Luego cerramos conexión:

conexion.Close();

Ahora en DISEÑO, usamos el elemento DataGridView, le ponemos nombre (dgvPokemons). Hacemos un evento Load en el Form y anotamos las siguiente líneas:

private void frmPokemons\_Load(object sender, EventArgs e)

{

PokemonNegocio negocio = new PokemonNegocio();

dgvPokemons.DataSource = negocio.listar();

}

Creamos una instancia “negocio” y a esta la asignamos los datos de dgvPokemons