

INTRODUCCIÓN:

La aplicación se ha implementado usando el framework .NET de Microsoft para el desarrollo y la tecnología ADO para el acceso a datos. Todo ello se ha realizado usando el lenguaje de programación orientado a objetos C#. El proyecto persigue principalmente el objetivo de realizar todo el ciclo de vida de la aplicación Web con el entorno orientado a objetos de Microsoft llamado .NET y con la tecnología ADO.NET para acceder a las bases de datos.

Para el desarrollo de la aplicación se ha utilizado la librería linq2db descargada desde NuGet para gestionar la conexión con la base de datos y poder gestionar las tablas como objetos.

CASO:

Se trata de desarrollar un sistema en .Net en el que se puedan dar de alta tres tipos de dispositivos: contadores de agua, de luz y gateways.

Donde como parámetros tienen:

Contador de agua: id, número de serie, marca, modelo

Contador de luz: id, número de serie, marca, modelo

Gateway: id, número de serie, marca, modelo, ip, puerto

ESTRUCTURA:

El código se divide en los siguientes archivos:

Los archivos .aspx donde está el código que define los Web Forms o interfaz gráfica de cada página. Los ficheros son: *about.aspx* y *default.aspx*.

Los archivos .aspx.cs donde está el código que implementa las funcionalidades de cada página. Los ficheros son: *about.aspx.cs* y *default.aspx.cs*.

Los archivos .cs que definen clases no vinculadas a ninguna página, se dividen en:

- Entidades de la base de datos: *Meter.cs* y *Gateway.cs*.
- Conexión con la base de datos: *Conexion.cs*
- Servicios que a su vez llaman a la conexión con la base de datos: *GateWayService.cs* y *MeterService.cs*.

El archivo Web.config en el que se guarda la configuración general de la aplicación web y en el que se ha añadido la configuración de la conexión con la base de datos MS Sql Server.

CASO DE USO:

El usuario selecciona en un desplegable el dispositivo que quiere dar de alta. Al seleccionar una u otra acción aparecerán o desaparecerán distintos datos a rellenar y pulsará el botón para guardar. En el caso de que alguno de los datos obligatorios no se haya rellenado aparecerá un alert con un aviso, al igual que en el caso de que este ya existiera.

DESPLIEGUE:

Para el despliegue necesitaremos lo siguiente instalado en nuestro equipo:

- Git Herramienta indispensable.
- Visual Studio.

El primer paso es localizar el repositorio a descargar y obtener el enlace. El botón *clone* nos provee del enlace que usaremos para clonar el repositorio. Desde el equipo donde queremos clonarlo, nos posicionamos en el directorio que queramos, e introducimos el siguiente comando:

- `git clone enlace`

Esto nos crea un clon local del repositorio remoto.

Ahora ya tenemos todo el proyecto correctamente descargado, falta importar las dependencias, para esto abrimos el proyecto usando Visual Studio. Para proceder ahora, tenemos que indicar a la solución que permita que Nuget maneje todas las dependencias, para esto debemos de hacer botón derecho en la solución y pulsar en la opción *Enable Nuget package Restore*.

Esta opción hace que Nuget descargue las dependencias correctas conforme las necesite, es decir, al compilar, cosa que vamos a hacer ahora para poder proseguir, para ello hacemos botón derecho en la solución y Build.

Esto compilará la solución por primera vez, pero lanzará un mensaje de error que podemos observar en la parte inferior. Esto es debido a que la primera compilación restablece los paquetes de NuGet, y en la segunda compilación es cuando realmente se descargan, por lo que vamos a compilar otra vez.

Y ya, por fin, podemos ejecutar nuestro proyecto correctamente.

Por último será necesario instalar Microsoft Sql Server y ejecutar el script de BBDD que se adjunta en el proyecto para crear la BBDD.