**Documentación UsersManagement**

Autor del crimen: Eduard Moret Ruz

**Una breve descripción**  
En este ejercicio pongo en practica mis habilidades como programador.El el contexto del ejercicio he decidido crear un proyecto con el frontend y el backend totalmente desacoplado para disponer de un mayor mantenimiento e independencia entre las partes del proyecto.

Las especificaciones técnicas son las siguientes:

Backend:

* Proyecto MVC WepAPI2
* Entity Framework 4.6.1
* Linq
* Arquitectura N-Capas (Presentación, Negocio, Acceso a datos)
* Desacoplamiento de dependencias con Ninject.
* Securización de la API con OAuth.
* Proyecto SQL para desplegar en una LocalDB
* Capa de Tests unitarios
* Patrón repositorio

Front-End

* Framework SPA: Angular 7
* Html5
* Css3
* TypeScript

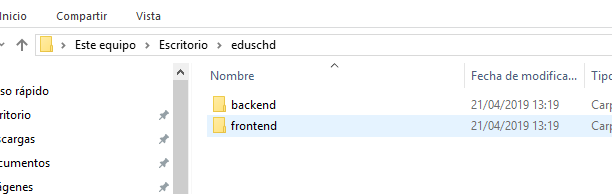
IDES utilizados:

* Visual Studo 2017
* Visual Studio Code

**Acceso al código fuente**

Para obtener el código fuente lo haremos en dos partes:

Yo personalmente obtaria por crear un directorio con el nombre del proyecto y dentro dos carpetas más. Una de ellas la nombraremos backend y la otra frontend como muestro en la imagen:

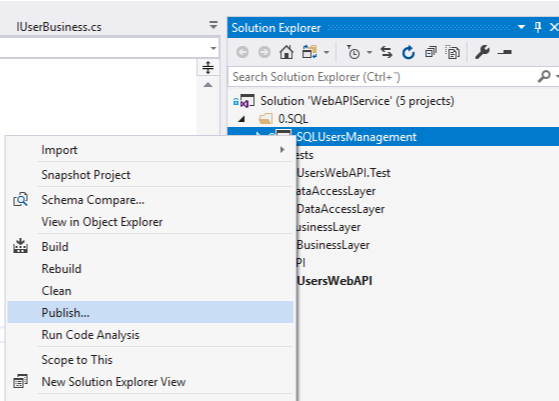


Entraremos dentro de la carpeta backend y mediante un ciente Git nos descargaremos el código del repositorio GitLab. Yo como costumbre utilizo tortoise git pero esto a gusto del consumidor..  
  
Hacemos clone del repositorio:

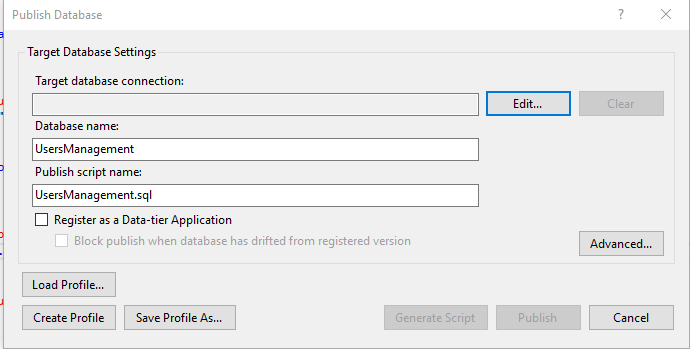
<https://gitlab.com/edward_catacrocker/webapi2-test.git>

**Despliegue del backend**

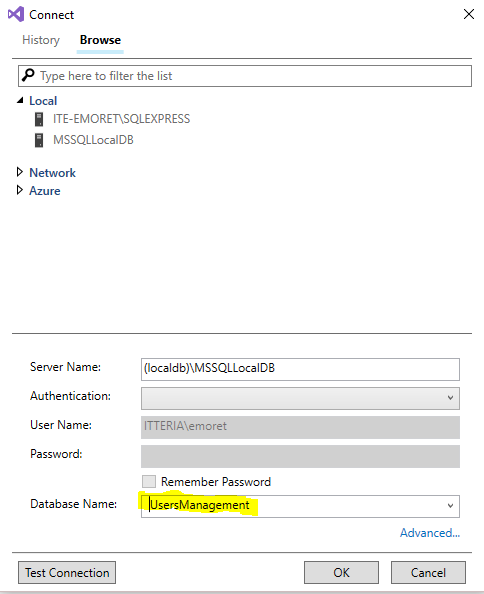
Una vez tenemos el código fuente a nuestra disposición seleccionamos el fichero **WebApiService.sln** i se nos abrirá Visual Studio . Veremos esta estructura y clicaremos con el botón derecho al proyecto **SQLUsersManagement** tal y como muestro en la siguiente imagen y le damos a publish:



Nos saldrá el asistente para desplegar la base de datos…

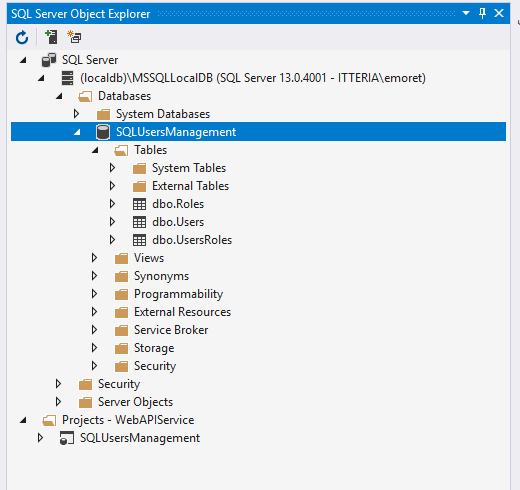


Escribimos UsersManagement en el campo Database name (importante) y le damos al botón Edit.  
  
Seleccionaremos browse local y la opción de LocalDB. Esto nos va a desplegar una base de datos en local sin necesitar gestionar la base de datos con un SQL Server Management Studio.



Nuestra base de datos la debemos nombrar **UsersManagement (es importante que se llame igual)** probaremos la conexión por si acaso con “Test Connection” i si ha ido bien seleccionaremos **OK**.

Ahora se no habrá desplegado una base de datos con los datos de los usuarios incluidos en local.  
  
Como veis ahora tenemos una base de datos desplegada y i la podremos gestionar desde el propio Visual Studio a través del **SQL Server Object Explorer.**



Desplegando la SPA

Angular es un framework que se levanta sobre un entorno Node.Js por lo tanto es importante que antes de seguir con los siguientes pasos instalemos Node en nuestra máquina. Lo podemos descargar aquí.

<https://nodejs.org/en/>

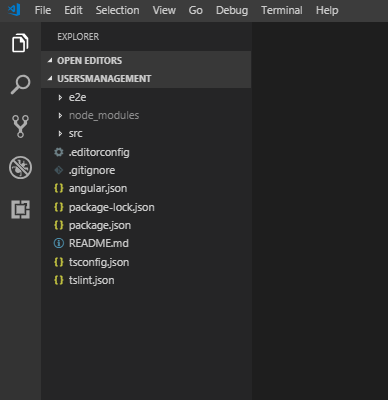
Yo recomiendo instalar la versión LTS(long term support) para un correcto funcionamiento…

Una vez tengamos el backend totalmente operativo tendremos que obtener el código fuente del proyecto frontend. Para ello, tal y i como hemos hecho con el paso anterior, hacemos un clone del proyecto a la carpeta FrontEnd que hemos creado inicialmente.

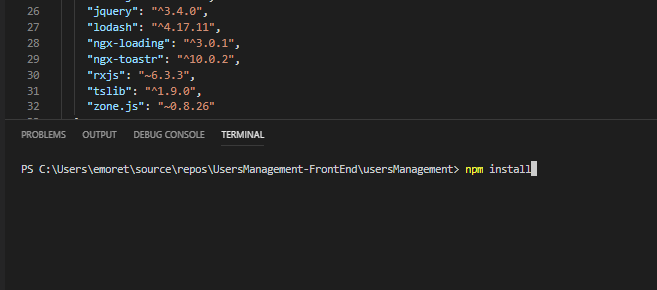
La url de gitlab es la siguiente:

<https://gitlab.com/edward_catacrocker/users-frontend.git>

Una vez tengamos descargado e instalado Visual Studio Code , abriremos el proyecto desde la raíz de la aplicación Angular. Ahora desde el IDE podremos visualizar el proyecto de esta manera:

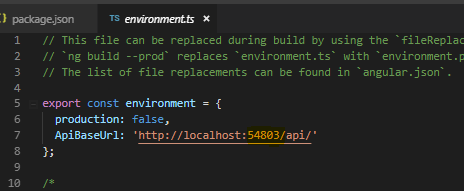


Instalando las dependencias

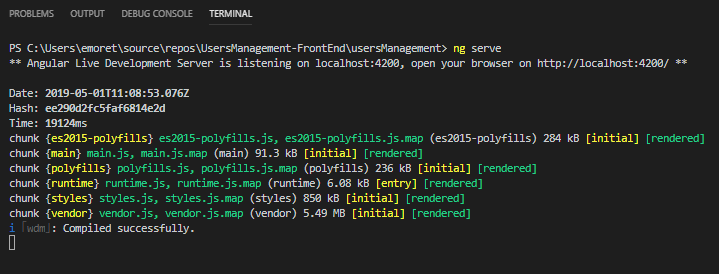
Para que nuestra aplicación angular funcione correctamente necesitaremos instalar las dependencias que necesita, las que tiene por defecto angular y las que he instalado yo. Como ya sabéis estas dependencias no vienen includidas en el directorio node\_modules pero si están declaradas en el fichero package.json. Para instalar estas dependencias lo haremos desde la ternimal que nos provee el IDE y escribiremos “npm install”. Esto leerá nuestro fichero package.json y descargará via npm(node package manager) las dependencias.  
  


Es posible que aparezca algún warning al acabar la instalación pero no va a crear ningún problema.

Una vez tengamos las dependencias instaladas navegaremos hasta el fichero environment.ts del proyecto. Escribimos Ctrl P i buscamos environment.ts. Una vez lo tenemos abierto nos aseguramos que el puerto en ApiBaseUrl apunta al mismo que nuestro backend. En mi caso lo tenia de esta forma:



Una vez llegamos a este punto podemos levantar nuestra aplicación angular a través de node. Para ello escribimos desde la misma ruta donde hemos **npm install** un **ng serve** o si no tuviéramos el intérprete de comandos de angular(angular cli) hacemos un **npm build** y luego un **npm start,** pero en principio tendríamos que tener instalado angular-cli como dependencia.

Si después de hacer el ng serve ha ido todo bien se nos informará esta información por pantalla que la compilación ha ido correctamente y el puerto dónde está escuchando la aplicación. En mi caso sobre el puerto 4200 de localhost.  
  


Abrimos el navegador que queramos (yo preferiblemente utilizo Chrome) y pongo la url: [http://localhost:4200](http://localhost:4200/)

Ahora una vez tenemos el frontend y el backend levantados accedemos a la url y se nos mostrará la pantalla de login.

Juego de usuarios:

User Password

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| edu |  |  |  |  |  |  |  |  | siberet |
| nuria |  |  |  |  |  |  |  |  | borrissol |
| gabri |  |  |  |  |  |  |  |  | farigola |
| manu |  |  |  |  |  |  |  |  | machupichu |
| Zigor |  |  |  |  |  |  |  |  | huhui |
| Donald Duck |  |  |  |  |  |  |  |  | hehei |
| Cheewbaka |  |  |  |  |  |  |  |  | hohoi |
|  |  |  |  |  |  |  |  |  |  |

**Nota importante:**

**Las constraseñas de esta aplicación no están encriptadas para favorecer así una cómoda manipulación de las mismas en base de datos.**

Salut!