CONFIGURACION DE LA APP EN IIS

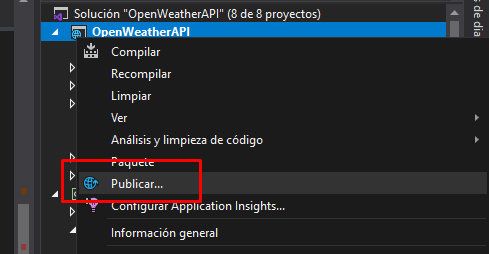
Se detallan los pasos para realizar configurar y probar la aplicación en un iis:

1-Activar IIS si el mismo no se encuentra activo:

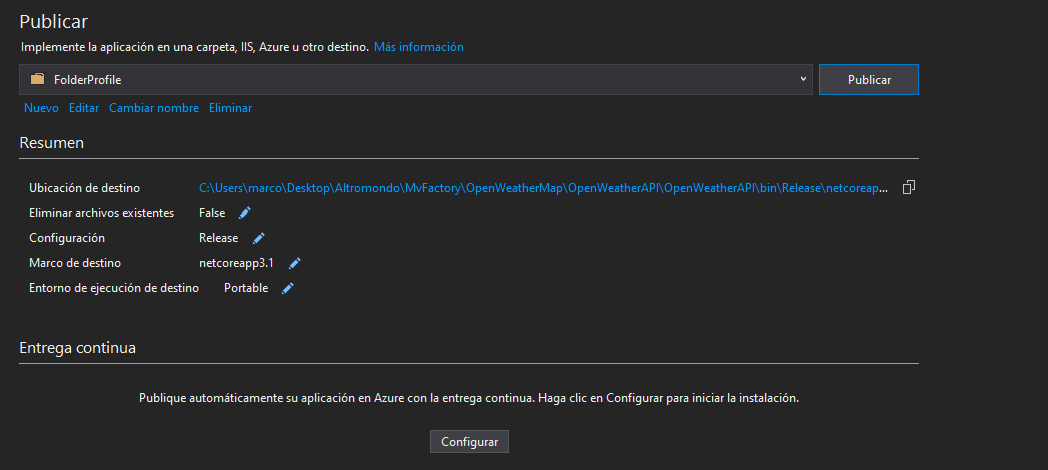
<https://enterprise.arcgis.com/es/web-adaptor/latest/install/iis/enable-iis-10-components-server.htm>

2-Generar el publish de la API de Net Core y el build de la aplicación angular que luego se dejaran en el directorio C:\inetpub\wwwroot\ para utilizar en el iis.

A-Generar el publish de la API de Net Core: en el visual studio nos paramos sobre el proyecto principal y hacemos botón derecho y click en la opción publish/publicar:

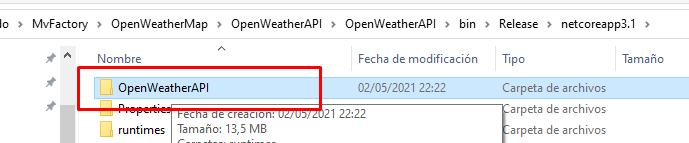


Luego de eso nos aparecerá una pantalla con configuraciones de la publicación:

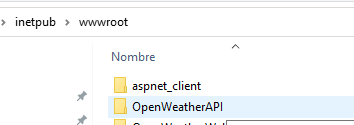


Seteamos la ubicación donde queremos que se generen los fuentes de nuestra publicación y dejamos los demás valores como están y hacemos click en publicar.

Una vez se generaron los archivos correspondientes, vamos a buscarlos al path donde se generaron:



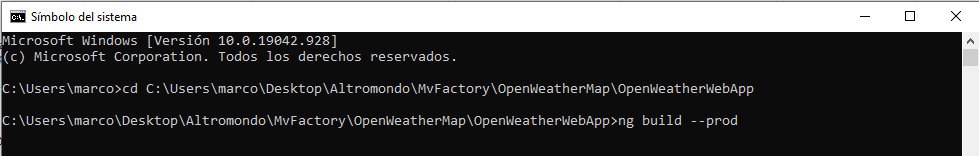
Copiamos la carpeta completa y la ponemos en el directorio C:\inetpub\wwwroot\



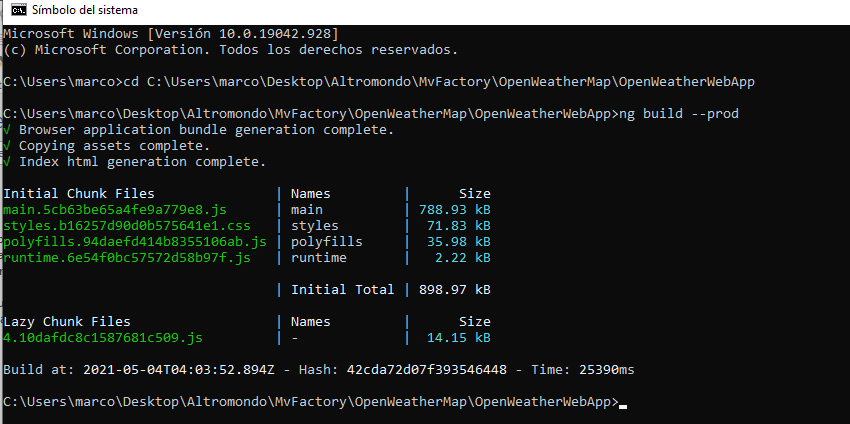
3-Generar el build de la aplicación angular:

Abrimos la cmd y nos ubicamos en la carpeta raíz de nuestro proyecto en angular

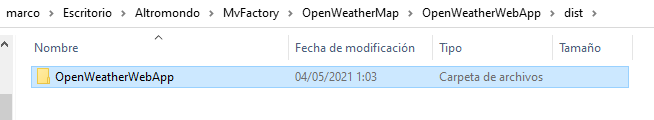
Y ejecutamos el comando ng build –prod



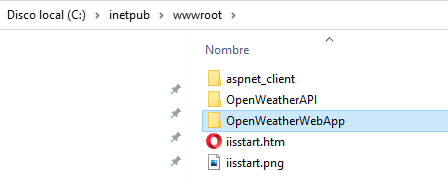
Una vez ejecutamos el comando se genera el compilado de la aplicación:



Buscamos los archivos en la carpeta dist, donde fueron generados:



Copiamos la misma y la llevamos al directorio C:\inetpub\wwwroot\



Generar un web.config con el siguiente contenido y dejarlo en la carpeta donde se encuentra la compilación del proyecto en angular.

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<system.webServer>

<rewrite>

<rules>

<rule name="Angular Routes" stopProcessing="true">

<match url=".\*" />

<conditions logicalGrouping="MatchAll">

<add input="{REQUEST\_FILENAME}" matchType="IsFile" negate="true" />

<add input="{REQUEST\_FILENAME}" matchType="IsDirectory" negate="true" />

</conditions>

<action type="Rewrite" url="./index.html" />

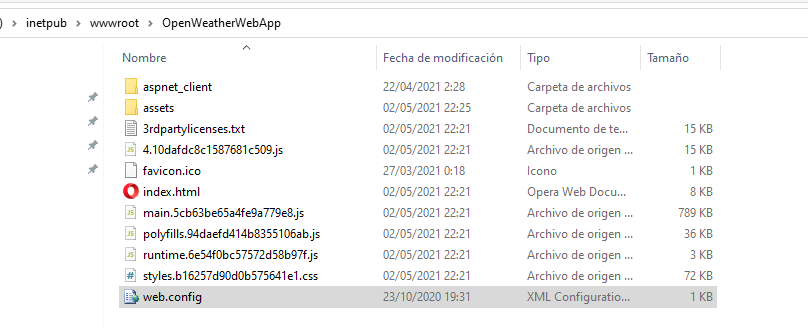
</rule>

</rules>

</rewrite>

</system.webServer>

</configuration>

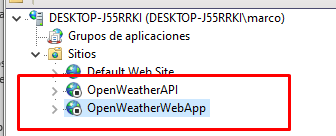


Para poder correr ambos proyectos en el iis previamente necesitamos tener instalado los siguientes programas:

-Microsoft .Net Core 3.0.3 – Windows Server Hosting

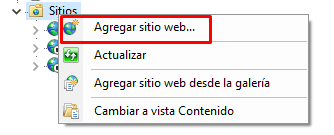
-Modulo URL Rewrite 2 de IIS(explicacion)

2-En el iis crear 2 sitios uno lo utilizaremos para la api de net core y otro para el sitio web en angular.

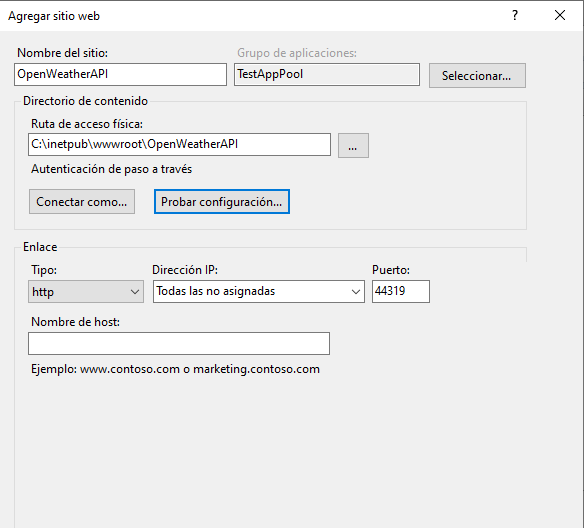


3-Configuracion del Sitio Web para la API de net core:

3-1.Boton derecho sobre Sitios y damos click en agregar Sitio Web:



3-2 Completamos los datos necesarios para configuración:



Nombre del Sitio: ponemos el nombre que querramos para nuestro sitio web

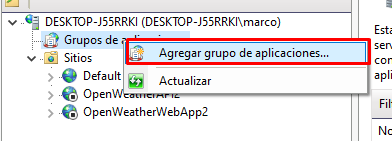
Ruta de acceso física: buscamos la carpeta que dejamos en C:\inetpub\wwwroot\ y la asociamos en esta opción.

Tipo: dejamos http puesto que no utilizaremos certificados SSL en nuestra aplicación.

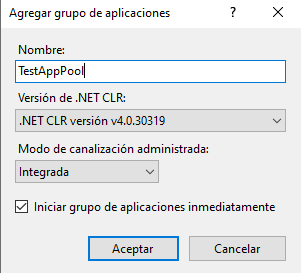
Puerto: elegimos uno libre.

Grupo de aplicaciones: Para evitar problemas de conexión de nuestra API con la base de datos y que estas dos puedan conectar seguimos los pasos tomados del pdf adjunto en la documentación fix\_asp\_net\_core\_login\_failed\_user\_iis\_pool.pdf

1. Creamos un usuario de Windows para este fin o bien podemos utilizar alguno del cual ya dispongamos con los permisos necesesarios.
2. Creamos un application pool

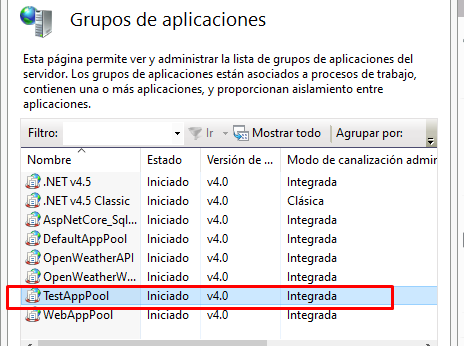


Lo configuramos de la siguiente forma:

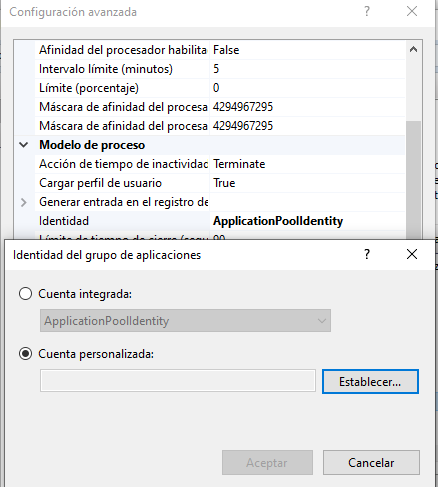


Clic en aceptar.

Hecho esto nos aparecerá nuestro app en la lista

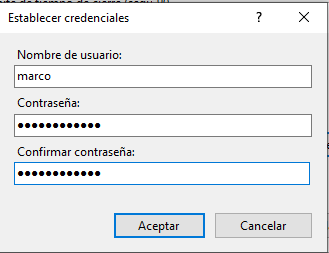


Luego vamos a configuración avanzada de nuestro AppPool que creamos y seleccionamos cuenta personalizada:



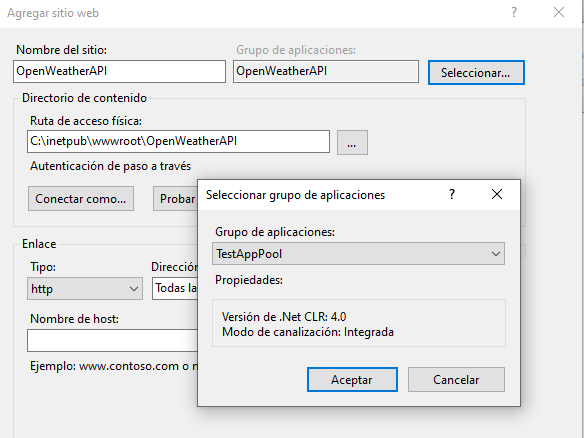
Hacemos click en establecer y se nos pedirá las credenciales de nuestra cuenta:

En este paso podemos usar el usuario de Windows que hayamos creado o bien utilizar alguno ya existente como mencionamos antes:

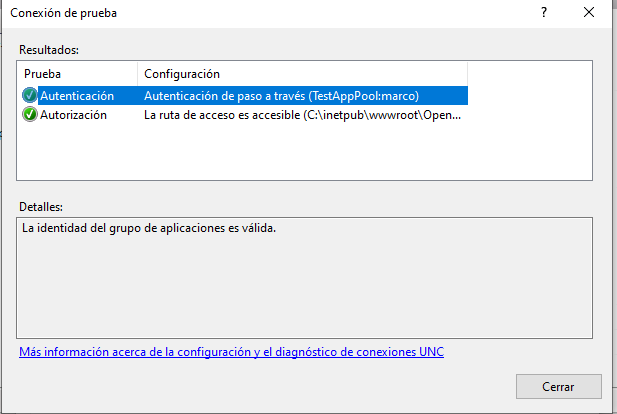


Damos clic en aceptar.

3-Ahora debemos asociar nuestro app pool a nuestro sitio web:



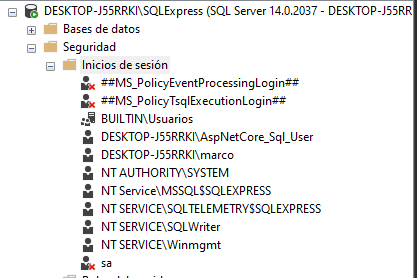
Para corroborar que esto funciona hacemos click en probar configuración, y deberíamos ver los siguientes mensajes:



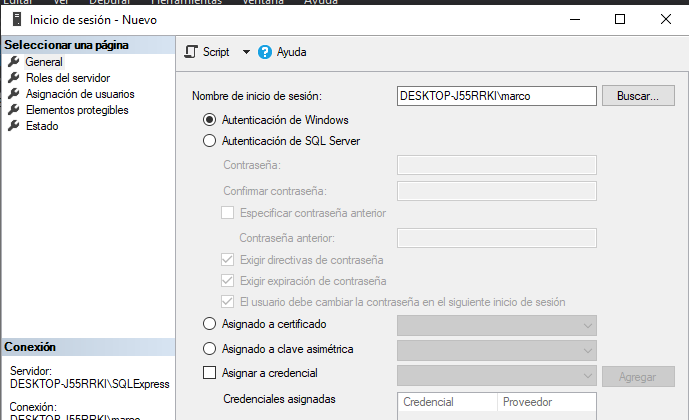
Cerramos y damos clic en aceptar y se creara nuestro sitio web.

4-Luego de esto debemos crear un usuario en Sql Server si es que ya ni lo tenemos creado.

Vamos a Sql Server->Seguridad->Inicios de Sesion

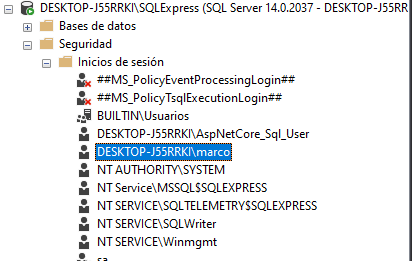


Aquí debemos crear un nuevo inicio de sesión utilizando el mismo usuario que usamos para el AppPool de nuestros sitios web

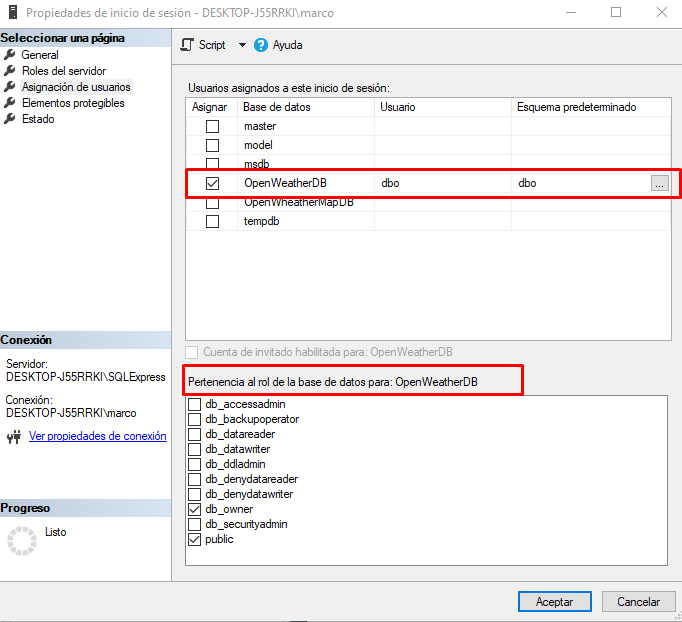


Damos clic en aceptar.

Y nos queda creado el inicio sesión:



Luego hacemos botón derecho propiedades sobre el mismo y damos clic en asignación de usuarios y seleccionamos la base de datos que hemos creado y los roles como se ve en la imagen siguiente:



Click en aceptar y reiniciamos el sitio web en el iis.

5-Configuracion del Sitio Web en Angular:

Debemos seguir mayoría de los pasas realizados anteriormente para configuración de la API de Net Core, a excepción de la creación del AppPool que ya lo teníamos creado de los pasos anteriores y el inicio de sesión en Sql Server

Pasos:

1-Creamos nuevo sitio en el iis

2-Le ponemos un nombre al sitio

3-Defininimos puerto

4-Seleccionamos como grupo de aplicaciones el TestAppPool definido anteriormente

5-Una vez completados todos los datos de nuestro sitio web damos clic aceptar y se creara el sitio web en angular.

6- Parametros de configuración de ambos proyectos:

A continuación se detallan lo parámetros utilizados en los archivos de configuración para ambos proyectos:

Sitio Web en angular:

App-config.json

“environment”: aquí se coloca la dirección de la api de net core con el que se comunicara nuestro sitio web.

API en Net Core:

Seccion “ConnectionString” 🡪”Default”: aquí se debe colocar el ConnectionString con el que nos conectaremos a la base de datos, utilizando autenticación de Windows Ej:

"Default": "Server = instancia; Database = Database; Trusted\_Connection = True;",

Seccion “ConnectionString” 🡪”OpenWeatherApp”: Es el la dirección del origen que se conectara a nuestra api de net core

"OpenWeatherWebApp": "http://localhost:4200"

Seccion “ApiSettings”🡪”ApiUrl”: aqui colocamos la direccion de la api del clima a la cual se consultan las condiciones climáticas, en id se colocara dinámicamente el id de la ciudad, en appid se colocara la api key que nos proporciona OpenWeatherMap

"ApiUrl": "http://api.openweathermap.org/data/2.5/weather?id={0}&appid={1}",

Seccion “ApiSettings”🡪”ApiKey”: Aquí colocaremos el key que nos proporciona OpenWeatherMap cuando nos registramos en el sitio.

"ApiKey": "code"

Seccion “JwtConfig”🡪SecretSecurityKey: Aquí colocamos la clave que utilizaremos para generar el token de autenticación.

Ej:

SecretSecurityKey: ”LoQueSea123456”

Seccion “JwtConfig”🡪IssuerAudience: Aquí colocaremos la dirección del emisor del token y el recimiente valido.

IssuerAudience: <http://localhot:54362>

Seccion “JwtConfig”🡪”TokenDuration”: tiemp de duracion del token en minutos