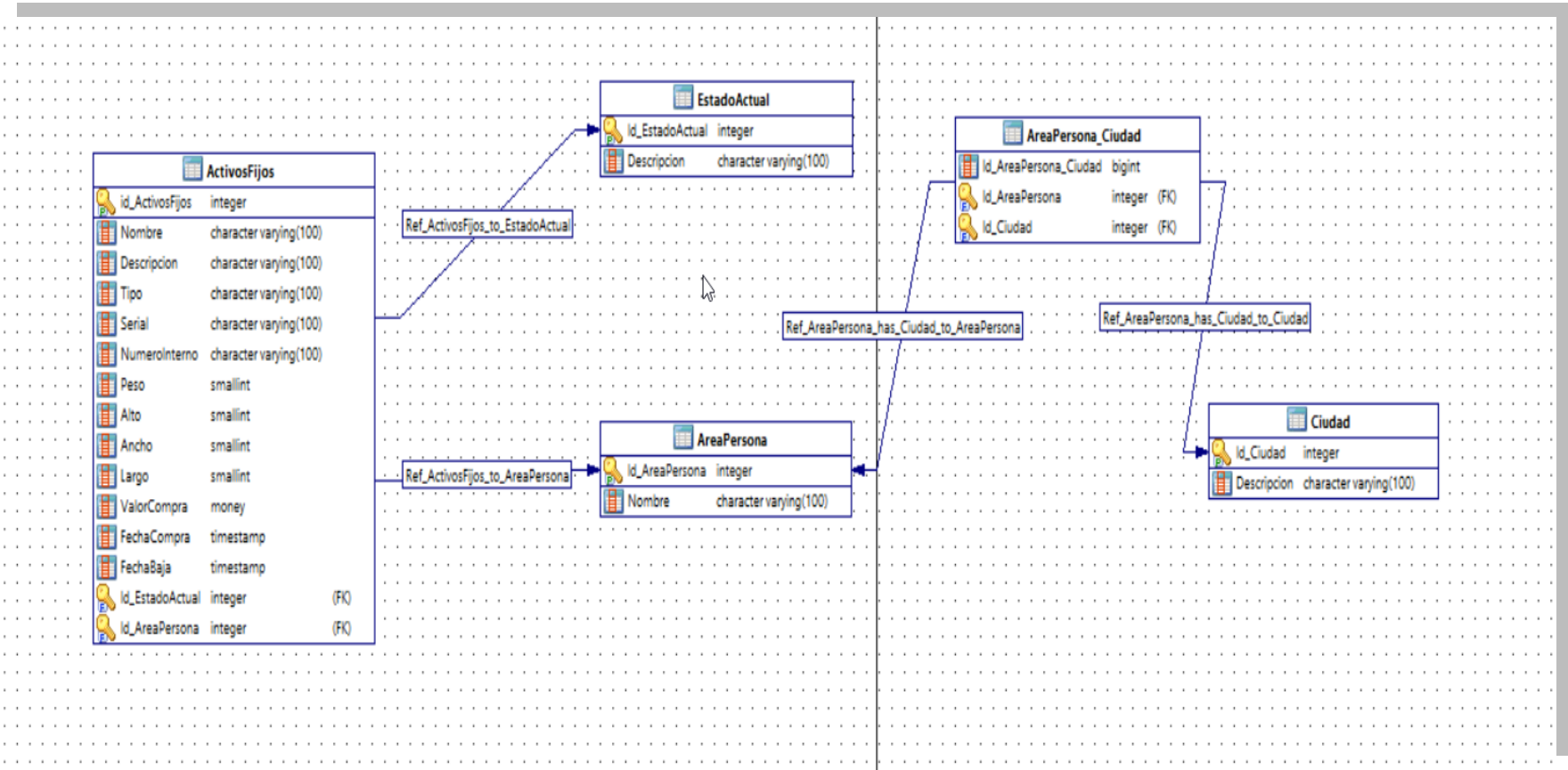
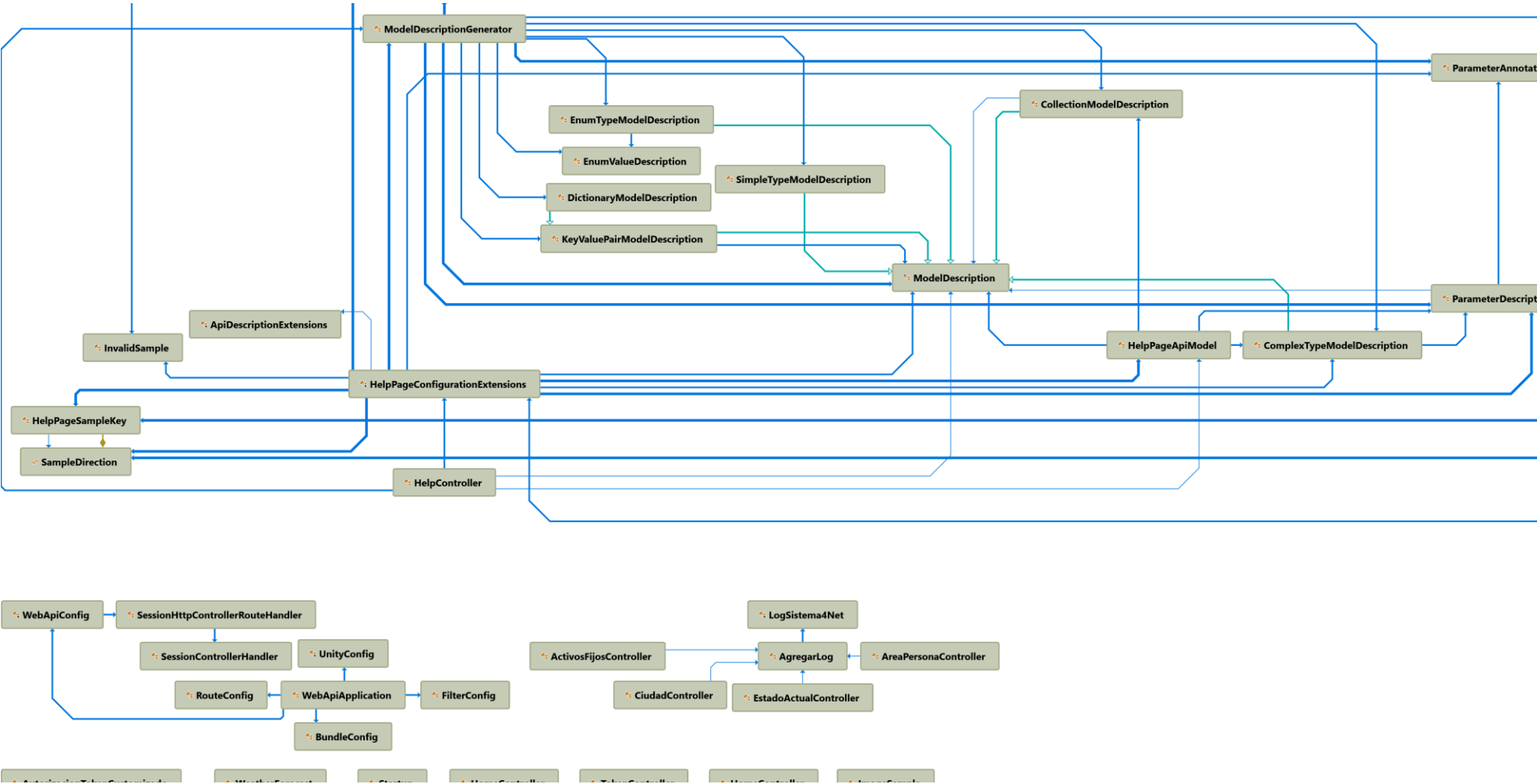


MODELO DE DATOS



ARQUITECTURA DE APLICACIÓN (web api)



Modelo de dependencias Arquitectura de componentes en N-Capas.

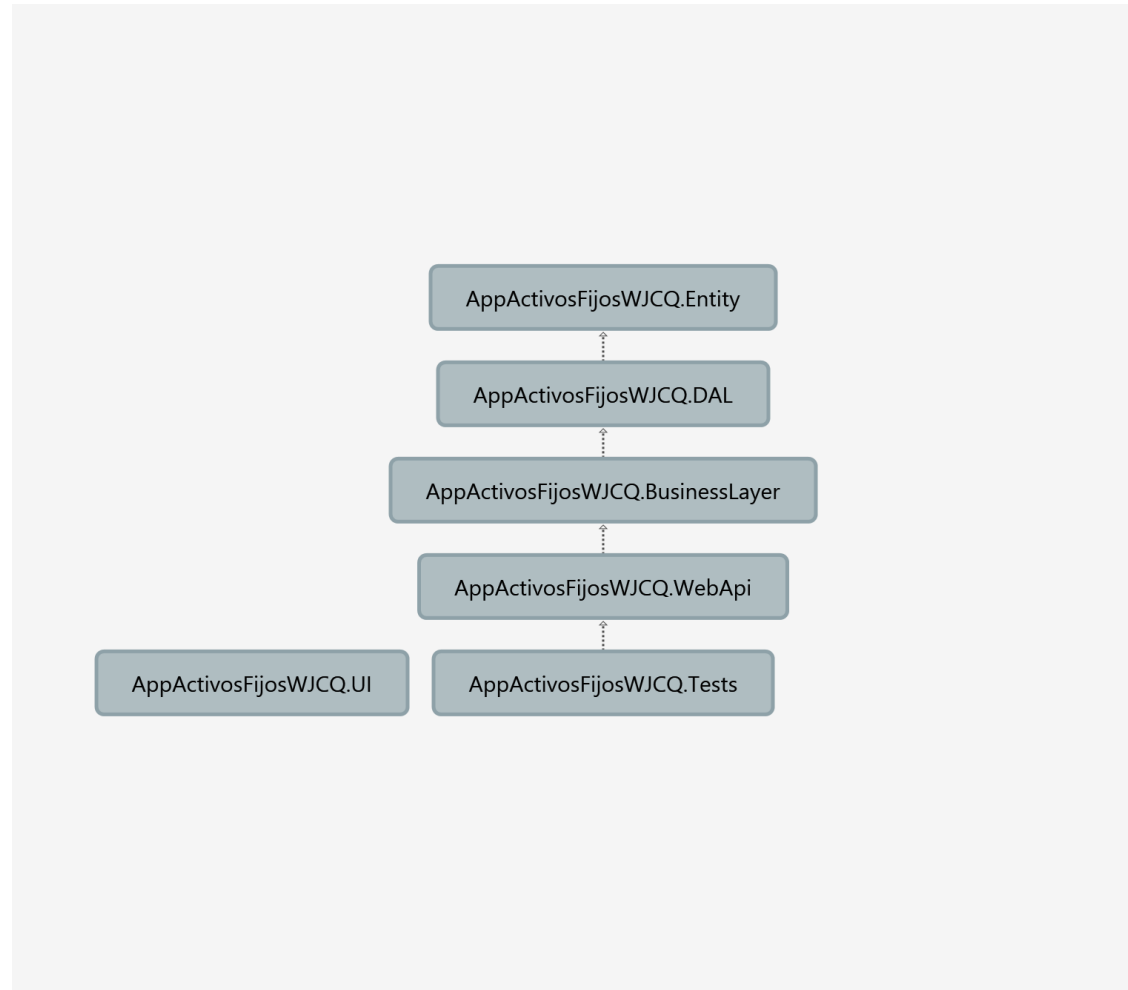
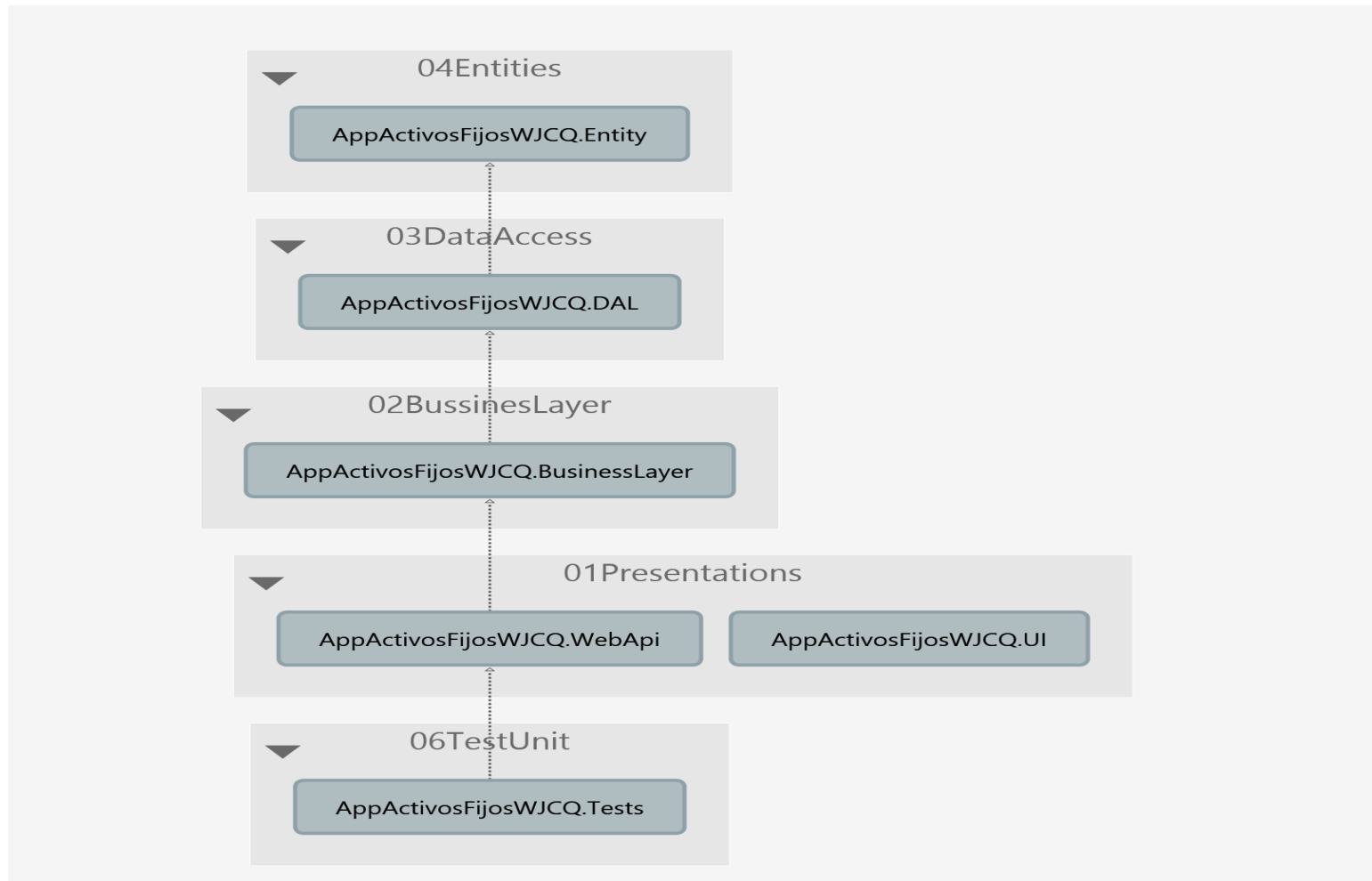
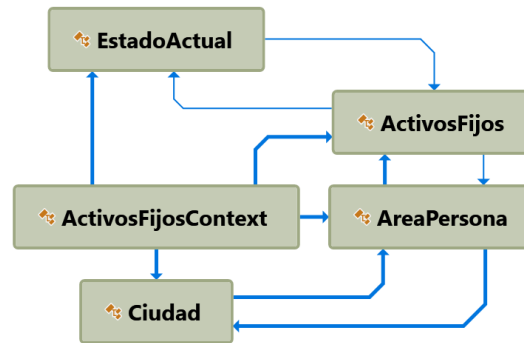


Diagrama de Artefactos.



Modelo de datos



## TECNOLOGIAS UTILIZADAS EN LA ARQUITECTURA.

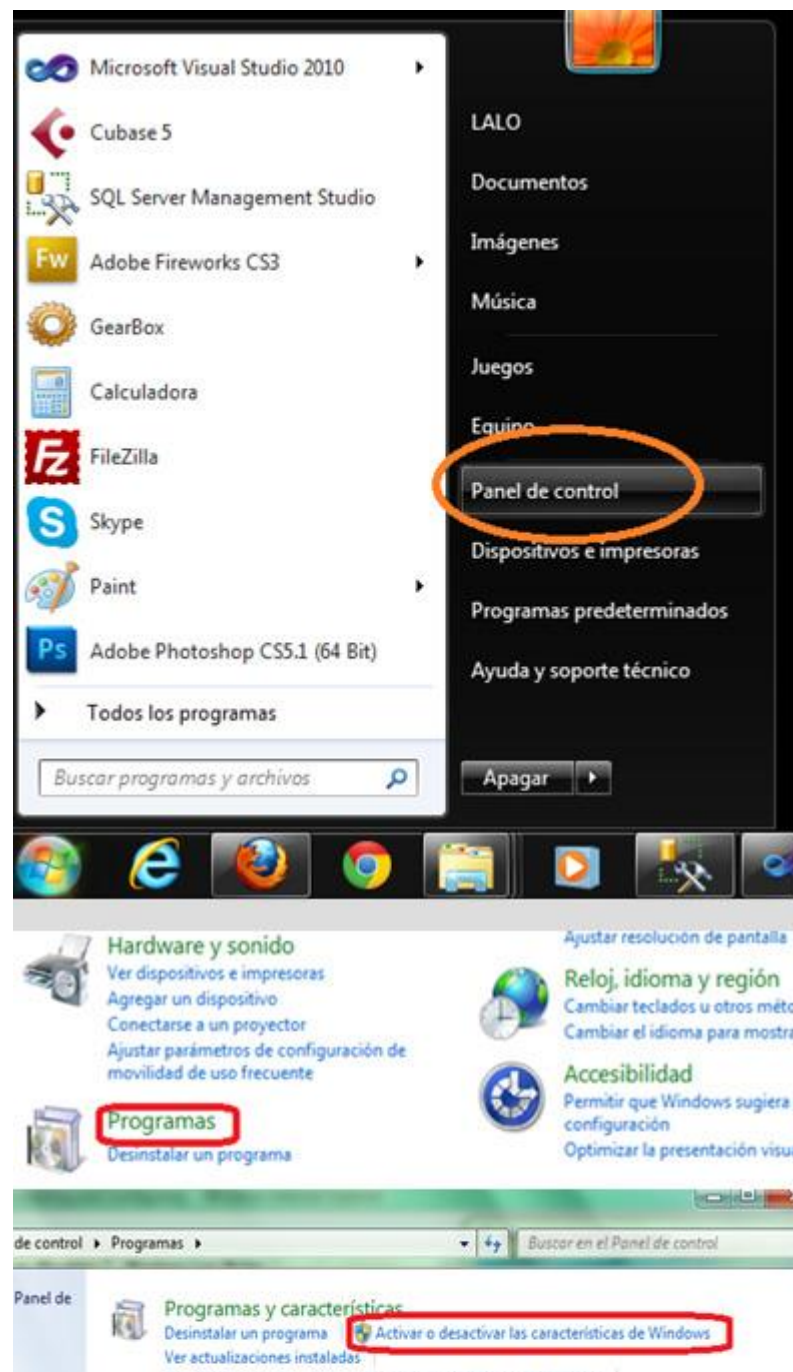
Se utilizaron tecnologías como:

- Unity para la inyección de dependencias con el fin de hacer que sea más mantenible el código y permitir realizar pruebas con mock que se orienten a TDD.
- Se utilizó CORS para el cross site de la aplicación
- Se utilizó Angular 5 con el fin de separar la capa de presentación a la de backend.(Se utilizaron tambien formularios reactivos y peticiones asíncronas)
- Se utilizó la base de datos postgresql y se utilizo el conector npgsql para c#
- Framework 4.6.1 de visual studio.
- Se utilizó sendel castel para la documentación
- Y para el web api se configuro el helper para su configuración su ruta es <http://localhost: 56993/> si se ejecuta en el ide.

## INSTRUCCIONES PARA GENERAR EL ARCHIVO DE DESPLIEGUE.

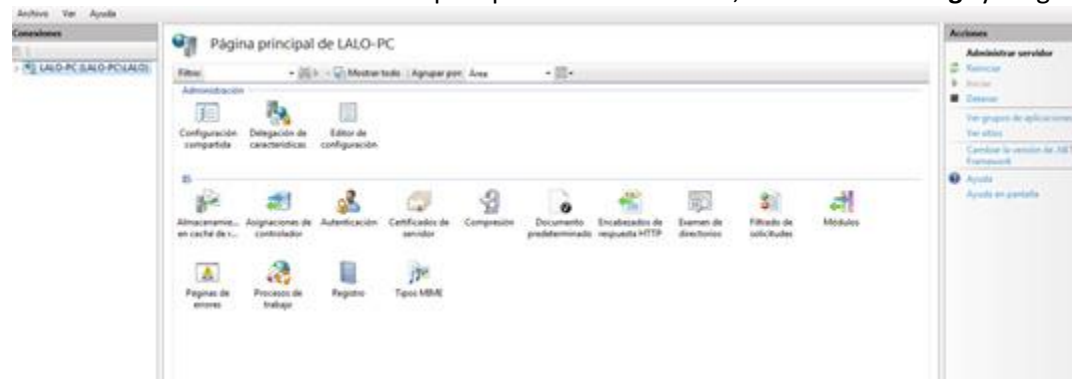
### PUBLICACIÓN DEL WEBAPI

1. Nos vamos al **Panel de control**.
2. Elegimos **Programas**.
3. Activar o desactivar **características de Windows**.
4. Habilitamos **Internet Information Services** y le damos **aceptar**.



### Configurando el IIS

Nos vamos ahora al panel de administración > sistema de seguridad > herramientas administrativas > y hacemos clic en Administrador de internet information services. Otra forma más rápida presionamos win + r, tecleamos **inetmgr** y luego enter.



Verificamos si está configurado correctamente entrando a nuestro browser y escribiendo **localhost**.



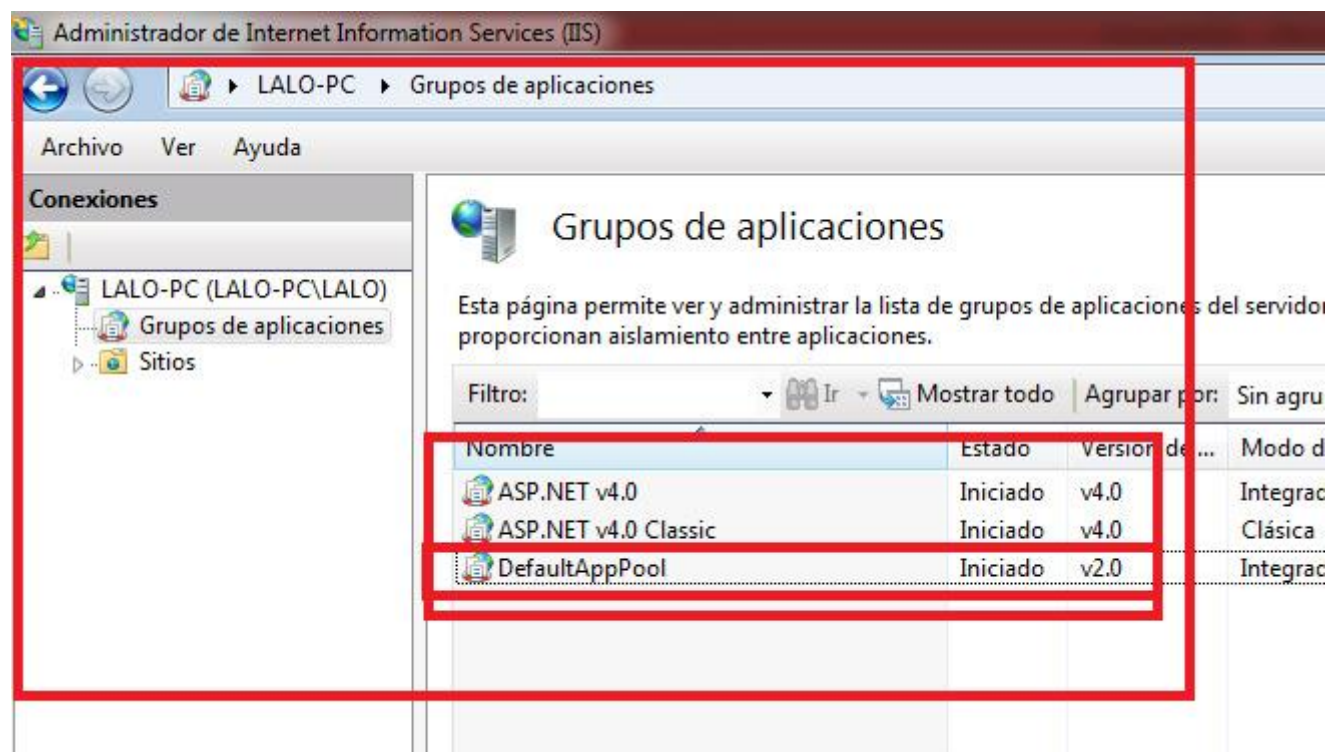


*Como les dije, yo lo hice desde Windows 7. Hoy en día, si usas un SO más moderno deberíamos ver otra página de inicio. La cosa es que salga algo como eso y no una página de error o 404.*

Ahora tenemos que **configurar el IIS** para que levante los aplicativos **ASP.NET**

- Entramos a la consola de Windows (win +r, escribir CMD)
- Copiamos esto `cd C:\Windows\Microsoft.NET\Framework\v4.0.30319\`
  - Copiamos `aspnet_regiis.exe -ir`

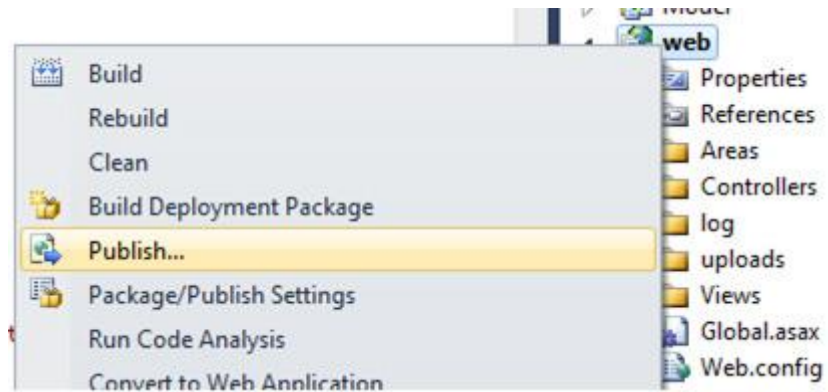
En el administrador del IIS (win + r, inetmgr) nos vamos a Grupo de aplicaciones.



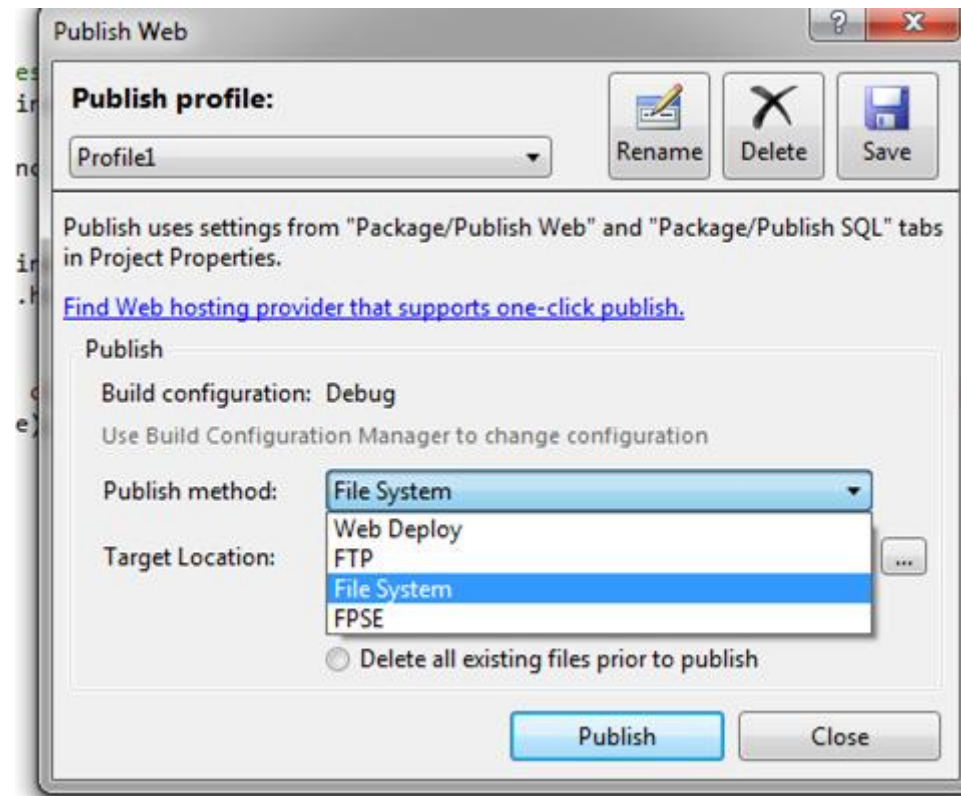
- Desplegamos grupos de aplicaciones
- Por defecto los aplicativos apuntan a DefaultAppPool, le damos doble click y cambiamos la versión del framework a la **4.0** o la más reciente.

Publicando nuestro proyecto en el IIS

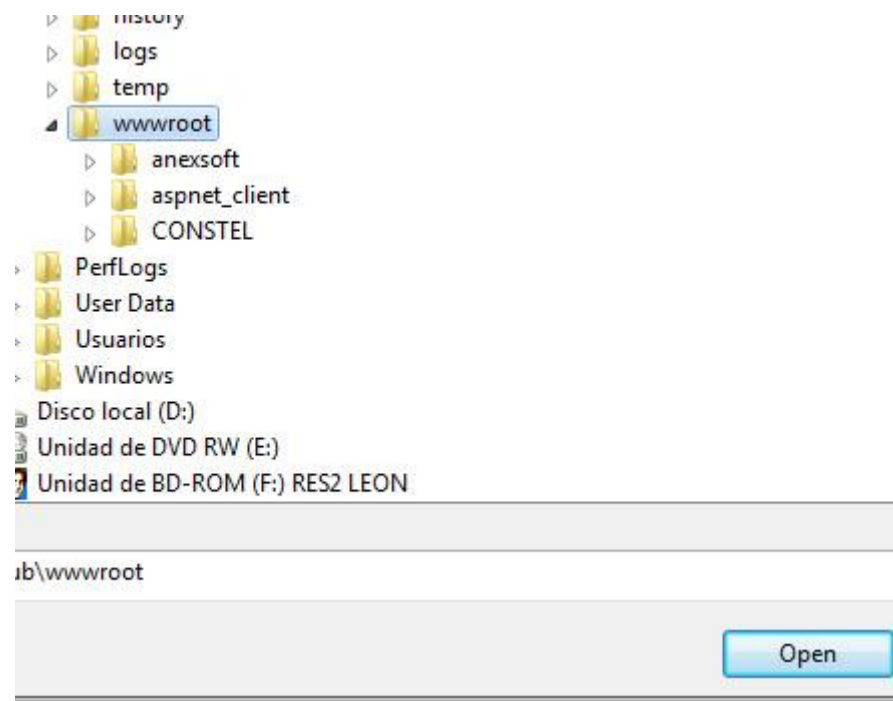
En nuestro proyecto, hacemos click derecho en la capa web y nos vamos a Publish.



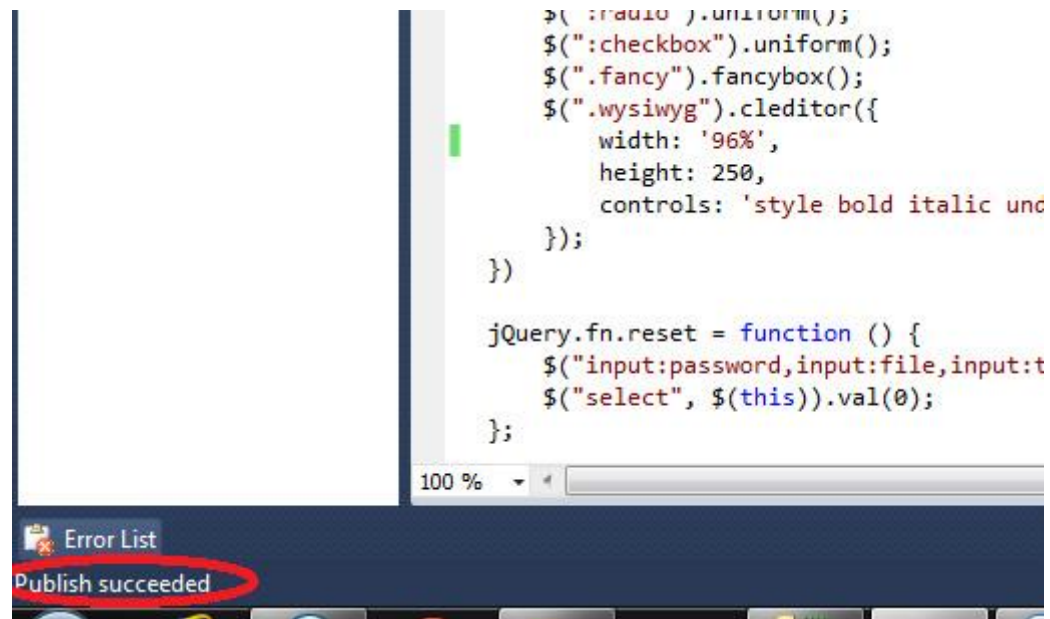
En publish method elegimos File System.



Luego elegimos la ruta donde va a estar el proyecto, en nuestro caso debe ser el inetpub/wwwroot (por defecto esta en el disco c:\inetpub\wwwroot).

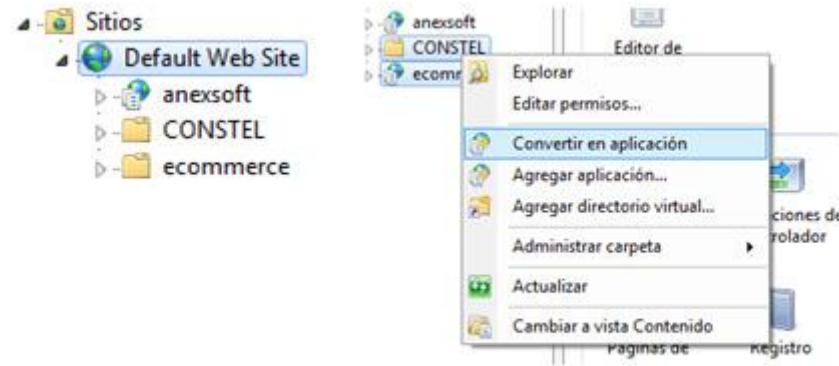


- Creamos una carpeta donde queremos guardar nuestro proyecto, y le damos open. Luego le damos clic en publish
  - Si todo salió bien veremos el siguiente mensaje en la parte inferior de nuestro visual studio.



Convirtiendo nuestra publicación en una aplicación

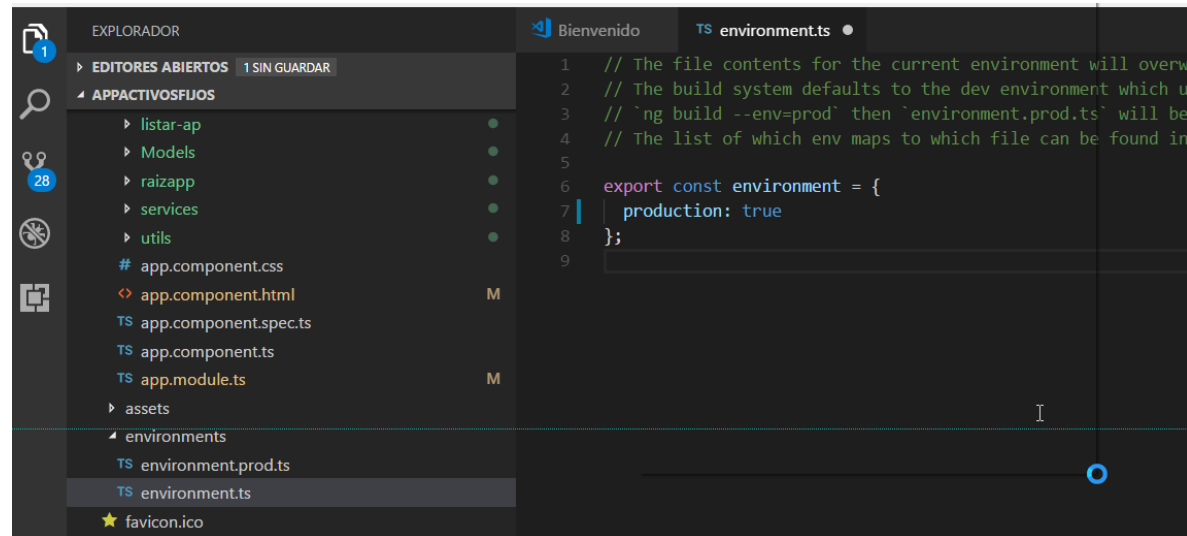
Regresamos al IIS y desplegamos las carpetas. Las que tengan icono de carpeta es que todavía no han sido convertidas en aplicación, le damos clic derecho y les decimos convertir aplicación.



## PUBLICACIÓN DE ANGULAR

Se configura la variable de producción en true

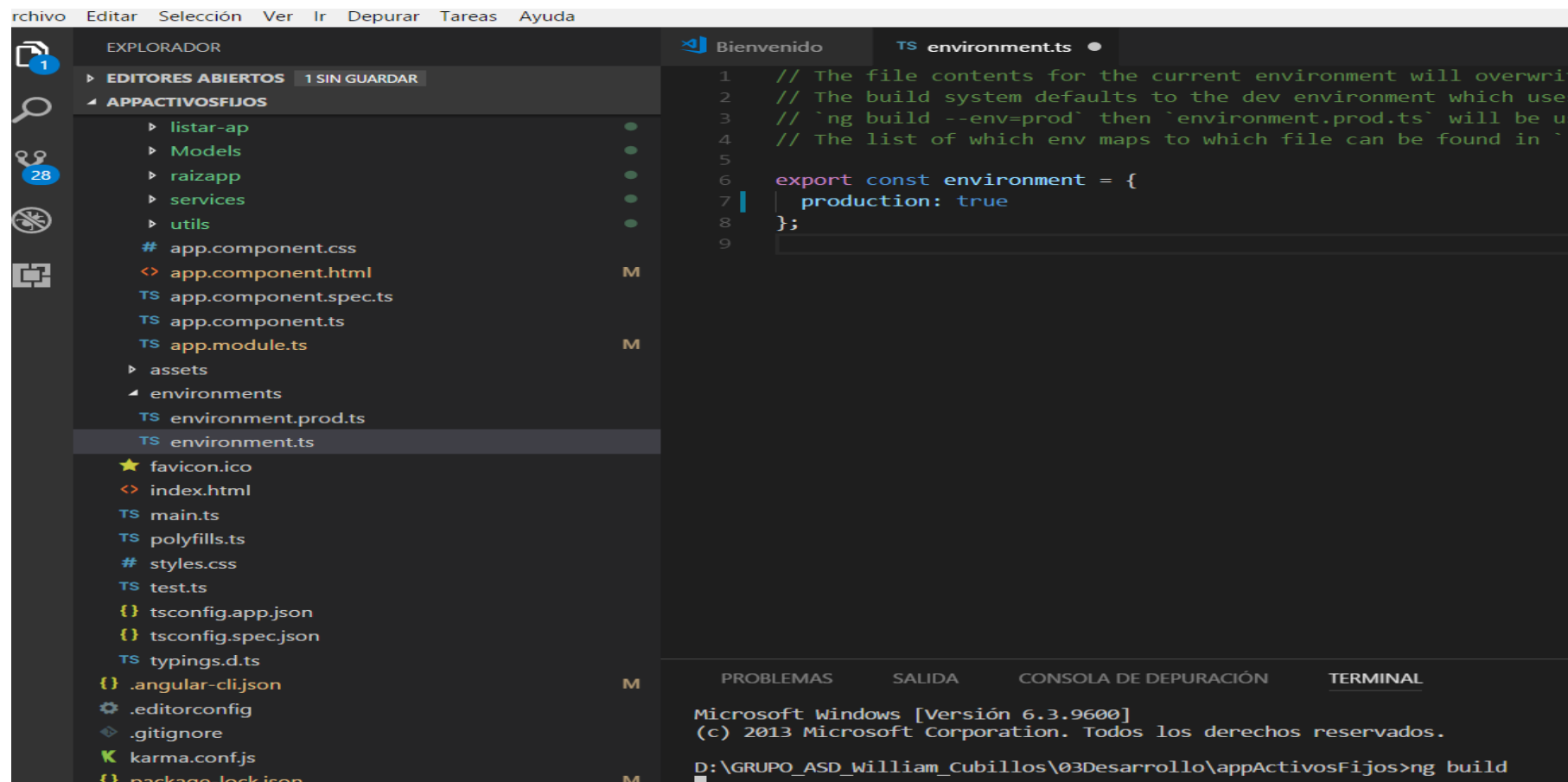
## PRUEBA WILLIAM CUBILLOS QUINTERO



Luego se abre la consola y se ubica en la carpeta del proyecto y se da el comando `ng build`


















## PRUEBA WILLIAM CUBILLOS QUINTERO



Esto genera una carpeta build

## PRUEBA WILLIAM CUBILLOS QUINTERO

 .git	31/01/2018 3:51	Carpeta de archivos	
 dist	31/01/2018 3:55	Carpeta de archivos	
 e2e	30/01/2018 11:05	Carpeta de archivos	
 node_modules	30/01/2018 20:47	Carpeta de archivos	
 src	30/01/2018 11:05	Carpeta de archivos	
 .angular-cli.json	30/01/2018 11:40	Archivo JSON	2 KB
 .editorconfig	30/01/2018 11:05	Archivo de origen ...	1 KB
 .gitignore	30/01/2018 11:05	Documento de tex...	1 KB
 karma.conf.js	30/01/2018 11:05	Archivo JavaScript	1 KB
 package.json	30/01/2018 20:47	Archivo JSON	2 KB
 package-lock.json	30/01/2018 20:47	Archivo JSON	474 KB
 protractor.conf.js	30/01/2018 11:05	Archivo JavaScript	1 KB
 README.md	30/01/2018 11:05	Archivo de origen ...	2 KB
 tsconfig.json	30/01/2018 11:05	Archivo JSON	1 KB
 tslint.json	30/01/2018 11:05	Archivo JSON	3 KB

Y esta carpeta hay que publicarla en el **ISS** como se mostró en los pasos anteriores