

4-6-2020

Ionic paso a paso II: Conceptos básicos



Sergio Maestre Pérez

ESCUELA DE INGENIERÍA DE GIPUZKOA – UPV/EHU

Título: Ionic paso a paso II: Conceptos básicos

Autor: Sergio Maestre Pérez

Fecha: 4 de Junio de 2020



Los contenidos de este documento están bajo la licencia Creative Commons de Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)

Las imágenes que aparecen en este documento provienen de los enlaces compartidos en el apartado: Más información

ÍNDICE

1. Introducción	Pág. 4
2. Ionic 3 vs. Ionic 4	Pág. 5
3. Crear un proyecto	Pág. 6
4. Estructura de un proyecto	Pág. 9
5. Generar nuevas funcionalidades	Pág. 12
6. Crear un servidor local	Pág. 15
7. Más información	Pág. 17

1) INTRODUCCIÓN

El objetivo de esta guía es explicar paso a paso todos los conceptos básicos que necesitamos adquirir para poder empezar a trabajar con Ionic Framework.

Antes de empezar es recomendable conocer el funcionamiento del CLI (Command Line Interface) de nuestro PC ya que en esta guía vamos a utilizar dicha herramienta en varias ocasiones.

En primer lugar, conoceremos las diferencias entre Ionic 3 e Ionic 4. A la hora de realizar un proyecto Ionic 3 nos dará más estabilidad aunque no dispondremos de las últimas novedades que nos ofrece Ionic 4.

En segundo lugar, conoceremos los pasos que hay que seguir para crear un proyecto, habiendo 3 opciones básicas: Proyecto en blanco, proyecto con menús laterales y proyecto con pestañas.

En tercer lugar, conoceremos la estructura que tiene un proyecto de Ionic 3, que es la version más estable y con la que se realizaran todos los proyectos explicados en esta guía y también la estructura que tiene un proyecto de Ionic 4, la cual nos ofrecerá las últimas novedades.

En cuarto lugar, conoceremos las distintas funcionalidades que podemos crear desde nuestro CLI (Command Line Interface) para nuestros proyectos.

En quinto y último lugar, conoceremos los pasos que hay que seguir para ejecutar un proyecto en un servidor local.

Al final de esta guía encontraréis distintos enlaces para poder profundizar más y de esta forma conocer un poco más los distintos conceptos básicos que necesitamos adquirir para poder empezar a trabajar con Ionic Framework.

2) IONIC 3 VS. IONIC 4



2.1 Ionic 4 y sus problemas de compatibilidad.

El lanzamiento de Ionic 4 por parte de Ionic Framework a finales del año 2018 trajo muchas novedades así como varios problemas de compatibilidad con los proyectos creados con Ionic 3.

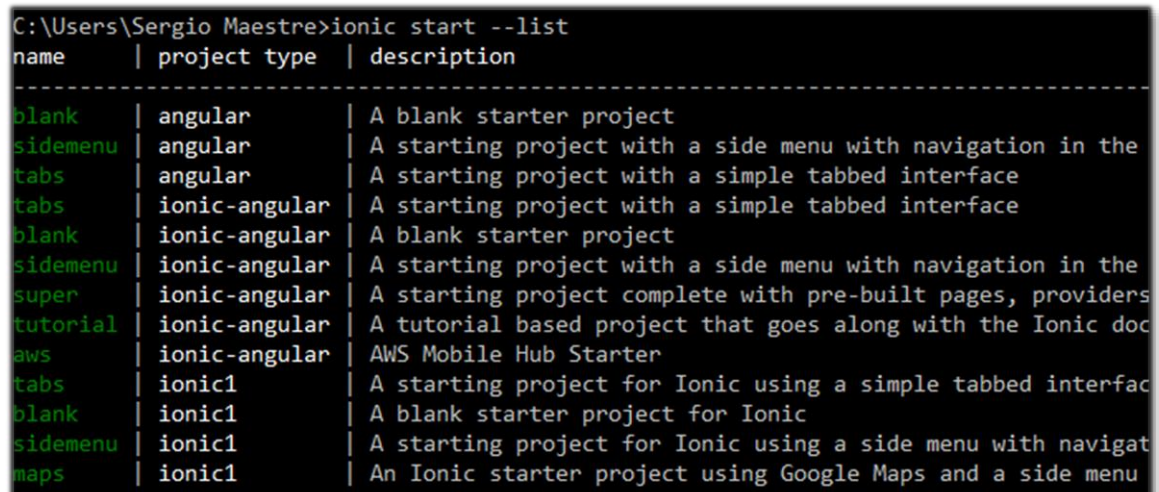
Todos los proyectos que aparecen en esta guía están hechos con Ionic 3 ya que la prioridad absoluta de esta guía es tener aplicaciones lo más estables posibles y hasta el momento, esto solo nos lo puede ofrecer Ionic 3.

2.2 ¿Cómo creamos un proyecto con Ionic 3 si tenemos instalado Ionic 4?

Escribimos en nuestro CLI el siguiente comando y pulsamos “enter” en nuestro teclado:

```
ionic start --list
```

Nos aparece la siguiente lista con los distintos tipos de proyectos que podemos crear con Ionic.



name	project type	description
blank	angular	A blank starter project
sidemenu	angular	A starting project with a side menu with navigation in the
tabs	angular	A starting project with a simple tabbed interface
tabs	ionic-angular	A starting project with a simple tabbed interface
blank	ionic-angular	A blank starter project
sidemenu	ionic-angular	A starting project with a side menu with navigation in the
super	ionic-angular	A starting project complete with pre-built pages, providers
tutorial	ionic-angular	A tutorial based project that goes along with the Ionic doc
aws	ionic-angular	AWS Mobile Hub Starter
tabs	ionic1	A starting project for Ionic using a simple tabbed interfac
blank	ionic1	A blank starter project for Ionic
sidemenu	ionic1	A starting project for Ionic using a side menu with navigat
maps	ionic1	An Ionic starter project using Google Maps and a side menu

Ilustración 1: Tipos de proyectos

La columna “Project type” nos indica en que versión de Ionic estamos trabajando.

- Ionic 4: angular
- Ionic 3: ionic-angular
- Ionic 1: ionic1

Por lo tanto, a la hora de crear un proyecto con Ionic 3 añadimos al final del comando “ionic start” el siguiente fragmento: “--type=ionic-angular”.

2.3 ¿Cómo migramos un proyecto de Ionic 3 a Ionic 4?

Dentro de la documentación oficial de Ionic Framework existe una guía para poder migrar un proyecto de Ionic 3 a Ionic 4. Para más información se puede consultar el siguiente enlace:

Migración de Ionic 3 a Ionic 4: <https://ionicframework.com/docs/building/migration>

3) CREAR UN PROYECTO



En primer lugar, ejecutamos nuestro CLI y nos colocamos en el directorio o carpeta donde queremos crear nuestro proyecto mediante el comando “cd”, del inglés “change directory” y que en castellano significa, cambiar de directorio.

Detrás de este comando, haciendo un espacio, escribimos el directorio donde queremos guardar nuestro proyecto y pulsamos “enter” en nuestro teclado.

```
C:\Users\Sergio Maestre>cd C:\Users\Sergio Maestre\Desktop\Mis proyectos  
  
C:\Users\Sergio Maestre\Desktop\Mis proyectos>
```

Ilustración 2: Crear un proyecto

En segundo lugar, escribimos el siguiente comando para crear un nuevo proyecto y hacemos “enter” en nuestro teclado:

`ionic start --type=ionic-angular`

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos>ionic start --type=ionic-angular  
  
Every great app needs a name!  
  
Please enter the full name of your app. You can change this at any time. To  
bypass this prompt next time, supply name, the first argument to ionic start.  
  
? Project name:
```

Ilustración 3: Nombre del proyecto

En tercer lugar, nuestro CLI nos pregunta por el nombre que queremos darle a nuestro proyecto, en nuestro caso le llamamos “Proyecto1” y hacemos “enter” en nuestro teclado:

`Proyecto1`

```
? Project name: Proyecto1

Let's pick the perfect starter template!

Starter templates are ready-to-go Ionic apps that come packed with everything
you need to build your app. To bypass this prompt next time, supply template,
the second argument to ionic start.

? Starter template: (Use arrow keys)
> tabs      | A starting project with a simple tabbed interface
blank      | A blank starter project
sidemenu   | A starting project with a side menu with navigation in the content
super      | A starting project complete with pre-built pages, providers and be
tutorial   | A tutorial based project that goes along with the Ionic documentat
aws        | AWS Mobile Hub Starter
```

Ilustración 4: Tipo de proyecto

En cuarto lugar, nuestro CLI nos pregunta qué tipo de proyecto queremos crear, habiendo 6 opciones:

- Proyecto con pestañas (tabs): Este proyecto consta de varias pestañas en blanco por las que se puede navegar.
- Proyecto en blanco (blank): Este proyecto carece de contenido alguno, tan solo una pantalla en blanco
- Proyecto con menús laterales (sidemenu): Este proyecto consta de varios menús laterales.
- Proyecto avanzado (super): Este proyecto consta de varias páginas pre-construidas, con un inicio de sesión y con otras funciones avanzadas de Ionic.
- Proyecto con tutorial (tutorial): Este proyecto consta de varias funciones que sirven a modo de tutorial.
- Proyecto con AWS (aws): Este proyecto consta de características especiales para trabajar con AWS, una plataforma de servicios de nube de Amazon.

Elegimos una de las 6 opciones en función del tipo de proyecto que queramos crear y pulsaremos “Enter” en nuestro teclado. En nuestro caso elegimos la segunda opción, es decir, un proyecto en blanco (blank).

Esperamos unos minutos a que el CLI cree nuestro proyecto.

Es posible que mientras nuestro proyecto se crea nos pregunten si queremos instalar Ionic Appflow SDK, a lo que respondemos que NO, ya que es una herramienta que no utilizamos en este proyecto.

```
? Install the free Ionic Appflow SDK and connect your app? (Y/n) n
```

Ilustración 5: Crear un proyecto

Una vez que acabe el proceso de creación nos aparece lo siguiente en nuestro CLI:

```
[INFO] Next Steps:

- Go to your newly created project: cd .\Proyecto1
- Run ionic serve within the app directory to see your app
- Build features and components: https://ion.link/scaffolding-docs
- Get Ionic DevApp for easy device testing: https://ion.link/devapp
```

Ilustración 6: Crear un proyecto

En quinto lugar, para poder empezar a trabajar en nuestro proyecto nos colocamos en el directorio o carpeta donde está ubicado nuestro proyecto mediante el comando “cd” del inglés “change directory” y que en castellano significa, cambiar de directorio:

cd Proyecto1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos>cd Proyecto1
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>
```

Ilustración 7: Crear un proyecto

En sexto y último lugar, ejecutamos el siguiente comando para añadir “cordova” y de esta forma darle a nuestro proyecto las propiedades necesarias para que este pueda ser híbrido:

ionic cordova prepare android

Es posible que mientras se esté ejecutando dicho comando nos pregunten si queremos instalar Platform Android, a lo que respondemos que SI.

```
? Platform android is not installed! Would you like to install it? (Y/n) y
```

Ilustración 8: Añadir cordova a un proyecto

Una vez que acabe la ejecución ya podemos empezar a trabajar en nuestro proyecto.

4) ESTRUCTURA DE UN PROYECTO



4.1 Carpeta raíz de un proyecto.

Si entramos en la carpeta que se genera al crear un proyecto vemos la siguiente estructura:

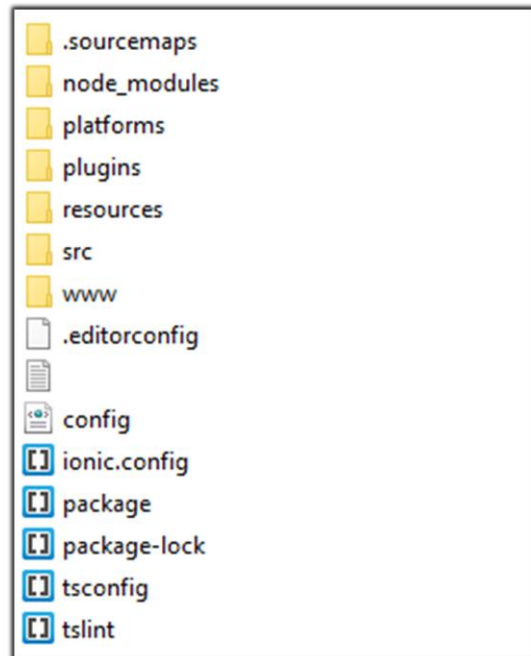


Ilustración 9: Estructura de un proyecto

En ella hay una serie de carpetas y archivos, y sus funciones son las siguientes:

Carpeta node_modules

Es una carpeta que se genera automáticamente al instalar las dependencias npm con el comando “npm install”. No tenemos que editar nada en esta carpeta.

Carpeta platforms

Es una carpeta que genera proyectos nativos para cada plataforma que hayamos añadido previamente. Si hemos añadido Android o iOS a nuestro proyecto mediante el comando “ionic cordova prepare” nos aparecerá una carpeta llamada Android o iOS que tendrán dentro las carpetas y archivos con la estructura de un proyecto nativo. No tenemos que editar nada en esta carpeta.

Tanto la carpeta **node_modules** como la carpeta **platforms** se actualizan cada vez que compilamos o ejecutamos un proyecto.

Carpeta plugins

Es una carpeta que contiene los plugins de Cordova que hayamos instalado previamente. Al instalar un plugin de Cordova se genera automáticamente dentro de esta carpeta. No tenemos que editar nada en esta carpeta.

Carpeta resources

Es una carpeta que contiene el icono y la “splash screen” de nuestra aplicación y que podemos generar reemplazando el archivo original por otro deseado y ejecutando después el comando “ionic cordova resources”.

Carpeta src

Es una carpeta que contiene todo el código de nuestro proyecto, es decir, es la carpeta más importante de nuestro proyecto ya que es donde constantemente tenemos que editar distintos archivos para cambiar el comportamiento y el aspecto de nuestra aplicación.

Carpeta www

Es una carpeta que se genera automáticamente y que contiene la version actual del código. Cada vez que editemos nuestro código que tenemos en la carpeta src esta carpeta se actualiza automáticamente. No tenemos que editar nada en esta carpeta.

Archivo config.xml

Es un archivo que contiene distintos parámetros que se utilizan cuando se construye un proyecto nativo a partir de un proyecto de Ionic. Aquí se indican los permisos especiales que necesiten nuestra aplicación y otras configuraciones que puedan ser necesarias. No tenemos que editar nada en este archivo.

Archivo ionic.config.json

Es un archivo que contiene información básica sobre la configuración de nuestro proyecto, se utiliza si subimos nuestro proyecto a la plataforma Ionic.io. No tenemos que editar nada en este archivo.

Archivo package.json

Es un archivo que contiene paquetes y dependencias de Node.js. No tenemos que editar nada en este archivo.

Archivo tsconfig.json y tslint.json

Son unos archivos que contienen información necesaria para compilar TypeScript. No tenemos que editar nada en estos archivos.

4.2 Carpeta src: El código de un proyecto.

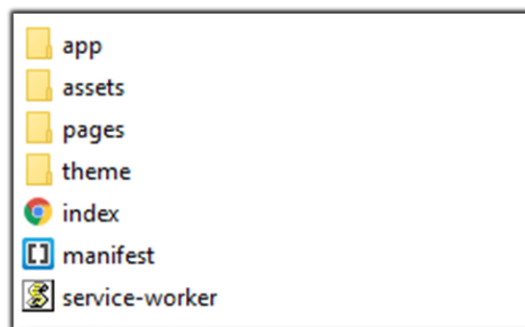


Ilustración 10: Estructura de un proyecto

Dentro de la carpeta src, encontramos 5 carpetas: App, Assets, Pages, Services y Theme.

App

Es una carpeta que tiene todos los archivos necesarios para ejecutar la aplicación y la estructura principal sobre la que construiremos la aplicación. Los archivos son:

- **app.component.ts:** Es un archivo que contiene el código del componente principal de la aplicación.
- **app.html:** Es un archivo donde se define la navegación y su raíz de la aplicación.
- **app.module.ts:** Es un archivo que contiene los distintos módulos de la aplicación.
- **app.scss:** Es un archivo que contiene los distintos estilos de la aplicación.
- **main.ts:** Es un archivo que se genera automáticamente y se encarga de la inicialización de la aplicación.

Pages

Es una carpeta que contiene todas las páginas de nuestro proyecto. Por cada página que creamos, dentro de esta carpeta aparecerá otra carpeta con el nombre de la página creada. Cada página tiene 3 archivos:

- **Pagina1.html:** Es el archivo html que contiene la plantilla de la página.
- **Pagina1.scss:** Es el archivo que contiene el archivo sass donde podemos modificar el estilo de los componentes de la página.
- **Pagina1.ts:** Es el archivo TypeScript que contiene toda la lógica de la página.
- **Pagina1.module.ts:** Es el archivo TypeScript que define el módulo de la página.

Theme

Es una carpeta que contiene los distintos temas y estilos y que hacen que nuestra aplicación sea mucho más fácil de personalizar.

Services

Es una carpeta que contiene todos los modelos y servicios para acceder a los datos remotos de nuestra aplicación.

Assets

Es una carpeta que contiene imágenes, datos de muestra y otros recursos que pueda necesitar nuestra aplicación.

5) GENERAR NUEVAS FUNCIONALIDADES



5.1 ¿Por qué es recomendable utilizar este comando?

El comando Ionic generate nos permite crear automáticamente distintas funciones que podemos utilizar en nuestra aplicación. De esta forma nos ahorramos tiempo y podemos desarrollar nuestro proyecto con mayor rapidez.

En primer lugar, ejecutamos nuestro CLI y nos colocamos en el directorio o carpeta de nuestro proyecto mediante el comando “cd”, del inglés “change directory” y que en castellano significa, cambiar de directorio.

```
C:\Users\Sergio Maestre>cd C:\Users\Sergio Maestre\Desktop\Mis proyectos
\Proyecto1
```

Ilustración 11: Generar nuevas funcionalidades

En segundo lugar, escribimos el siguiente comando para generar una nueva funcionalidad a nuestro proyecto y hacemos “enter” en nuestro teclado:

ionic generate

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate
? What would you like to generate: (Use arrow keys)
> component
  directive
  page
  pipe
  provider
```

Ilustración 12: Generar nuevas funcionalidades

Podemos generar las siguientes funcionalidades:

5.2 ¿Cómo generamos un Component (Componente)?

Un component es una pieza de código que podemos usar en cualquier parte de nuestra aplicación. Para generar un componente ejecutamos el comando:

ionic generate component Componente1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate
component Componente1
[OK] Generated a component named Componente1!
```

Ilustración 13: Generar un componente

Dentro de la carpeta src/components se genera la carpeta componente1 y el archivo components.module.ts. Dentro de la carpeta componente1 se generan otros 3 archivos.



Ilustración 14: Generar un componente



Ilustración 15: Generar un componente

5.3 ¿Cómo generamos un Directive (Modificador de atributos)?

Un directive es un modificador de atributos. Para generar un directive ejecutamos el comando:

ionic generate directive Directive1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate directive Directive1  
[OK] Generated a directive named Directive1!
```

Ilustración 16: Generar un modificador de atributos

Dentro de la carpeta src/directives se genera la carpeta directive1 y el archivo directives.module.ts. Dentro de directive1 se genera el archivo directive1.ts.

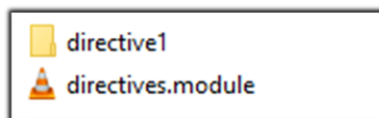


Ilustración 17: Generar un modificador de atributos

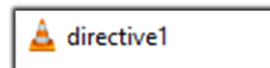


Ilustración 18: Generar un modificador de atributos

5.4 ¿Cómo generamos una Page (Página)?

Para crear una página ejecutamos el comando:

ionic generate page Pagina1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate page Pagina1  
[OK] Generated a page named Pagina1!
```

Ilustración 19: Generar una página

Dentro de la carpeta src/pages, se genera la carpeta pagina1 junto a la ya existente carpeta home, que se genera automáticamente al crear un proyecto. Dentro de la carpeta pagina1 se generan otros 4 archivos.

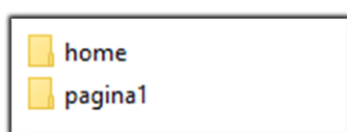


Ilustración 20: Generar una página

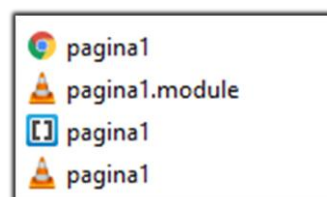


Ilustración 21: Generar una pagina

5.5 ¿Cómo generamos un Pipe (Transformador de datos)?

Un pipe es un transformador de datos. Por ejemplo: Mostrar textos en mayúsculas, formatos de fecha o reloj, etc. Para generar un provider ejecutamos el comando:

ionic generate pipe Pipe1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate pipe Pipe1
[OK] Generated a pipe named Pipe1!
```

Ilustración 22: Generar un transformador de datos

Dentro de la carpeta src/pipes, se genera la carpeta pipe1 y el archivo pipe.module. Dentro de la carpeta pipe se genera el archivo pipe.ts.

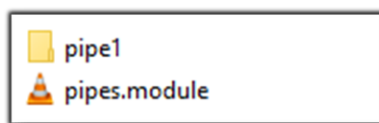


Ilustración 22: Generar un transformador de datos

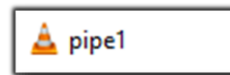


Ilustración 23: Generar un transformador de datos

5.6 ¿Cómo generamos un Provider (Proveedor de datos)?

Un provider es un servicio que se encarga de manipular los datos de almacenamiento en la nube. Para generar un provider ejecutamos el comando:

ionic generate provider Proveedor1

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic generate provider Proveedor1
[OK] Generated a provider named Proveedor1!
```

Ilustración 23: Generar un proveedor de datos

Dentro de la carpeta src/provider, se genera la carpeta proveedor1. Dentro de la carpeta proveedor1 se genera el archivo provider1.ts.



Ilustración 24: Generar un proveedor de datos



Ilustración 25: Generar un proveedor de datos

6) CREAR UN SERVIDOR LOCAL



En primer lugar, ejecutamos nuestro CLI y nos colocamos en el directorio o carpeta de nuestro proyecto mediante el comando “cd”, del inglés “change directory” y que en castellano significa, cambiar de directorio.

```
C:\Users\Sergio Maestre>cd C:\Users\Sergio Maestre\Desktop\Mis proyectos
\Proyecto1
```

Ilustración 26: Crear un servidor local

En segundo lugar, escribimos el siguiente comando para crear un servidor local de nuestro proyecto y hacemos “enter” en nuestro teclado:

ionic serve

```
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>ionic serve
> ionic-app-scripts serve --address 0.0.0.0 --port 8100 --livereload-
port 35729 --dev-logger-port 53703 --nobrowser
[app-scripts] [00:49:21] ionic-app-scripts 3.2.1
[app-scripts] [00:49:21] watch started ...
[app-scripts] [00:49:21] build dev started ...
[app-scripts] [00:49:21] clean started ...
[app-scripts] [00:49:21] clean finished in 7 ms
[app-scripts] [00:49:21] copy started ...
[app-scripts] [00:49:21] deeplinks started ...
[app-scripts] [00:49:21] deeplinks finished in 25 ms
[app-scripts] [00:49:21] transpile started ...
[app-scripts] [00:49:25] transpile finished in 4.32 s
[app-scripts] [00:49:25] preprocess started ...
[app-scripts] [00:49:25] preprocess finished in less than 1 ms
[app-scripts] [00:49:25] webpack started ...
[app-scripts] [00:49:26] copy finished in 4.64 s
[app-scripts] [00:49:29] webpack finished in 3.46 s
[app-scripts] [00:49:29] sass started ...
[app-scripts] [00:49:30] sass finished in 1.25 s
[app-scripts] [00:49:30] postprocess started ...
[app-scripts] [00:49:30] postprocess finished in 7 ms
[app-scripts] [00:49:30] lint started ...
[app-scripts] [00:49:30] build dev finished in 9.25 s
[app-scripts] [00:49:30] watch ready in 9.37 s
```

Ilustración 27: Crear un servidor local

Esperamos unos minutos a que el CLI cree nuestro servidor local. Una vez que acabe este proceso, se abre nuestro navegador y podemos observar nuestra aplicación.

La aplicación debería de ejecutarse sin problema alguno ya que no hemos editado el código de la misma. En caso de error, nuestro CLI nos daría un mensaje de error.

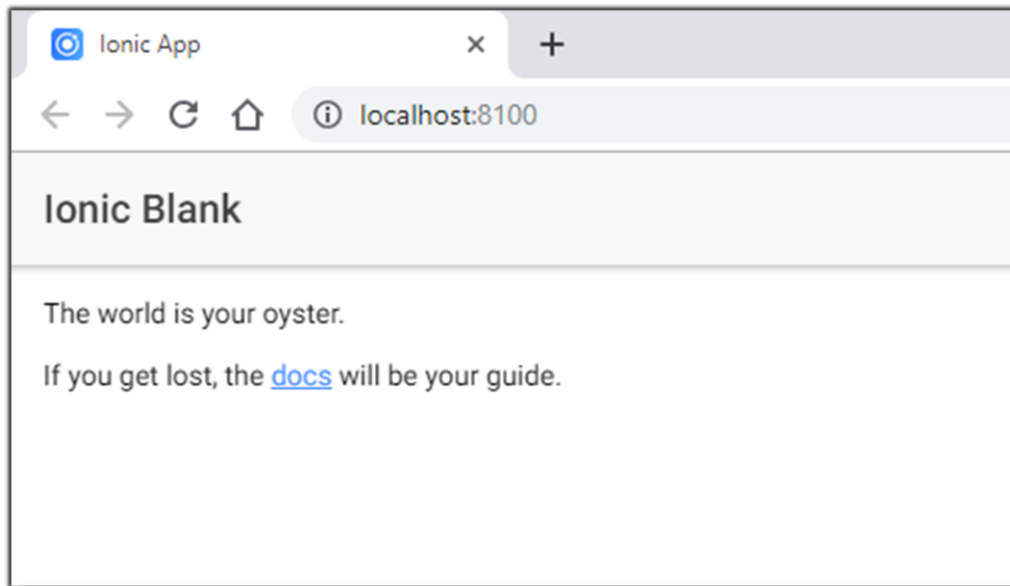


Ilustración 28: Servidor local visto desde el navegador web

Para salir de la aplicación, volvemos a nuestro CLI y hacemos “Ctrl+C” en nuestro teclado.

```
[INFO] Development server running!  
  
Local: http://localhost:8100  
External: http://192.168.0.16:8100  
DevApp: Proyecto1@8100 on LAPTOP-59DTBK8V  
  
Use Ctrl+C to quit this process  
  
[INFO] Browser window opened to http://localhost:8100!  
  
[app-scripts] [00:49:34] lint finished in 3.86 s
```

Ilustración 29: Crear un servidor local

Al realizar esta acción nuestro CLI nos pregunta si queremos cerrar el servidor local que hemos creado, a lo que respondemos que SI.

```
¿Desea terminar el trabajo por lotes (S/N)? s  
C:\Users\Sergio Maestre\Desktop\Mis proyectos\Proyecto1>
```

Ilustración 30: Crear un servidor local

7) MÁS INFORMACIÓN:

Para más información se pueden consultar los siguientes enlaces:

Página principal Ionic Framework: <https://ionicframework.com/>

Documentación Ionic Framework: <https://ionicframework.com/docs>

Documentación Ionic 3: <https://ionicframework.com/docs/v3/>

Comandos Ionic 3: <https://ionicframework.com/docs/v3/cli/commands.html>

Migración de Ionic 3 a Ionic 4: <https://ionicframework.com/docs/building/migration>

Estructura de un proyecto:

- <https://ionicframework.com/docs/v3/cli/projects.html>
- <https://ionicthemes.com/tutorials/about/ionic-application-structure>

Generar nuevas funcionalidades:

- <https://ionicframework.com/docs/building/scaffolding#generating-new-features>
- <https://blog.ng-classroom.com/blog/tips/ionic-generator/>

Página principal Ionic Themes: <https://ionicthemes.com/>

Tutoriales Ionic Themes: <https://ionicthemes.com/tutorials>

Página principal NG Classroom: <https://blog.ng-classroom.com/blog/>

Blog NG Classroom: <https://blog.ng-classroom.com/blog/ionic2/>

Página principal Stack Overflow: <https://stackoverflow.com/>

Foro Stack Overflow: <https://stackoverflow.com/questions/>

Selección de la licencia: <https://creativecommons.org/choose/>

