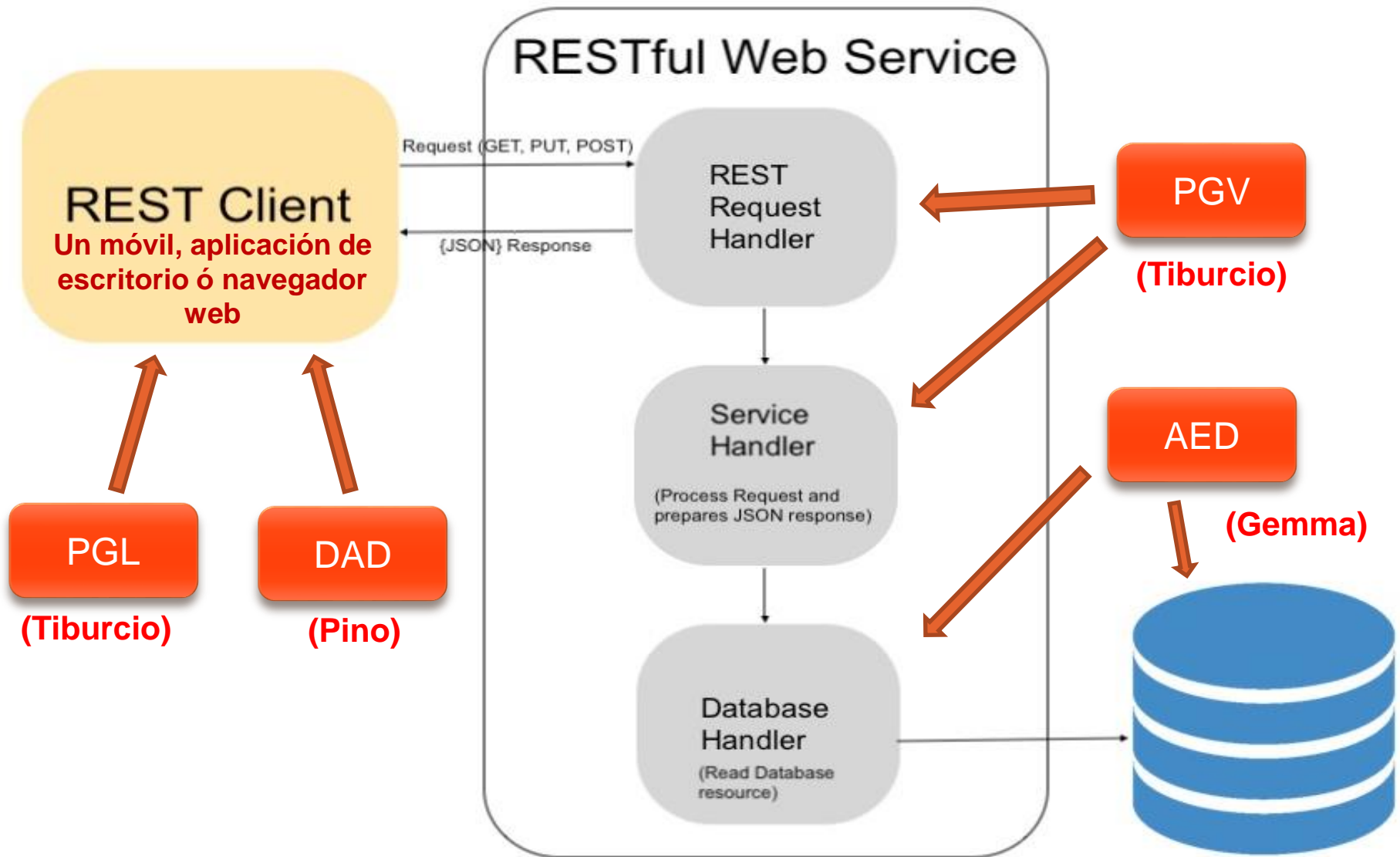


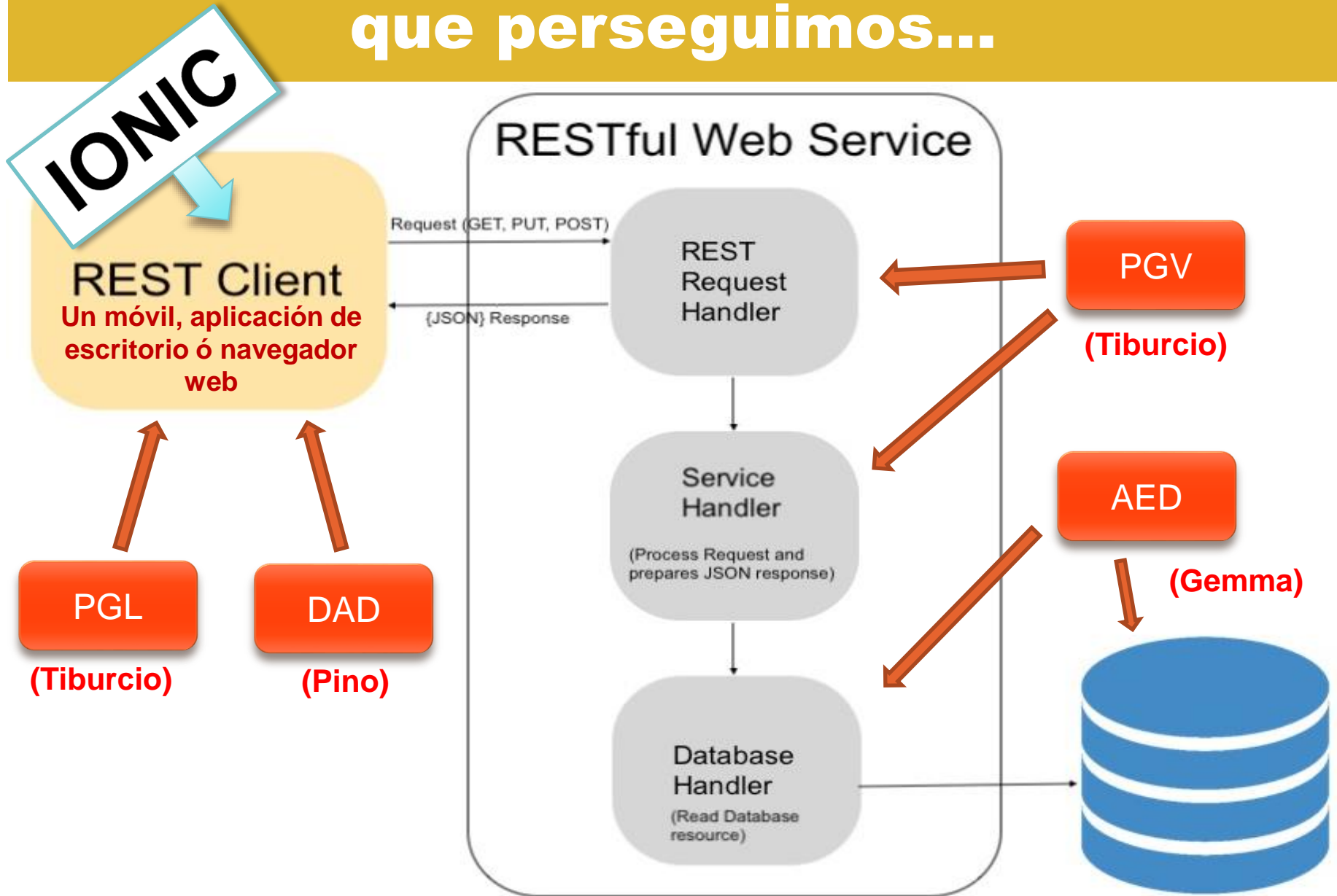
Introducción a IONIC (Usando Angular)

Resumen de pasos basado en la página oficial de Ionic: <https://ionicframework.com/docs>

No perdamos nunca la visión global que perseguimos...



No perdamos nunca la visión global que perseguimos...



Hybrid vs Native

Aquí se explica bien:

<https://go.ionicframework.com/hybrid-vs-native-guide>

(descárgate el libro)



Android
Studio



Al principio sólo se podía usar el conjunto de herramientas propias de cada dispositivo:

- Android, programando en Java (o ahora en Kotlin) con el IDE Android Studio.
- iOS, programando en Swift con el IDE Xcode.

Hybrid vs Native

La principal desventaja de las apps nativas es que hay que programar desde 0, para que tu proyecto funcione en cada una de las plataformas.

Con las apps nativas hay que programar al menos 3 veces el mismo proyecto



Hybrid vs Native

Las apps híbridas siguen el principio “**code once, run anywhere**”

- Se programa con JS, HTML y CSS.
- El código corre en un web view (un navegador que ocupa toda la pantalla)
- Este webview se incrusta en un envoltorio nativo (native wrapper) propio de cada dispositivo.
- Dicho envolverio nativo es realmente el que se ejecuta en el dispositivo real.



Hybrid vs Native

La principal desventaja de las apps nativas es que hay que programar desde 0, para que tu proyecto funcione en cada una de las plataformas.



Las apps híbridas sin embargo:

- Siguen el principio “**code once, run anywhere**”.
- En Ionic se programa en **HTML, CSS** y **JavaScript**.
- A continuación utilizando **cordova** o **capacitor** se puede crear una versión para **Android, iOS** o la **Web**.
- El código en realidad se ejecuta en un **webview** (navegador de tamaño de ventana completa) que se incrusta en una “envoltorio nativo” Android o iOS que permite a la App comportarse como una app nativa.
- El acceso a los recursos nativos del dispositivo se realiza mediante **plugins** (por ejemplo para la cámara).

¿Qué más hay en la viña del señor?

Native apps



Hybrid apps

Web apps

PWAs
(Progressive Web Apps)

- Utilizan las herramientas de desarrollo (SDK) de cada plataforma. Hoy en día Android e iOS.
- Se distribuye por medio de los repositorios como Google play (Android), o App Store (iOS).
- Hay que instalar la App para usarla.

¿Qué más hay en la viña del señor?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)



- Se programa con JS, HTML y CSS.
- El código corre en un web view (un navegador que ocupa toda la pantalla)
- Este webview se incrusta en un envoltorio nativo (native wrapper) propio de cada dispositivo.
- Dicho envoltorio nativo es realmente el que se ejecuta en el dispositivo real.
- Las Apps híbridas también pueden generar una app web con el mismo código.

¿Qué más hay en la viña del señor?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)



- Se ejecuta en un navegador.
- Por tanto funciona en cualquier dispositivo con navegador e Internet.
- Si no hay cobertura no funciona.
- No hay que instalar nada salvo el navegador.

¿Qué más hay en la viña del señor?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)



- Es una solución intermedia entre una web app y una app nativa.
- Es decir: tiene una interfaz similar a la de una App Nativa. Funcionan sin conexión a Internet. Permite el envío de Notificaciones Push. Se actualiza de forma automática. Es Instalable. Se pueden compartir a través de una URL.

¿Qué es Ionic?

Ionic Framework is an open source mobile UI toolkit for building high quality, cross-platform native and web app experiences



Básicamente:

- Es **open source**.
- Es un conjunto de herramientas (**toolkit**).
- Permite crear experiencias de **apps multiplataforma** tanto **nativas** (android y ios), como **web**.
- Se programa en **HTML, CSS y JavaScript**.
- Es “**framework agnostic**”. Se puede usar **React, Angular, Vue** ó incluso con JavaScript sin más (**JavaScript Vanilla**)
- **Code once, run anywhere.**

Al lío... vamos a hacer nuestra primera App con Ionic...

Pasos previos

En la página oficial de Node:

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

Descarga e instala la versión LTS de NodeJS



16.13.0 LTS

Recomendado para la mayoría

17.1.0 Actual

Últimas características

Al ío... vamos a hacer nuestra primera App con Ionic...

Java 17 available now

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

JDK 17 will receive updates under these terms, until at least September 2024.

Java SE Development Kit 17.0.1 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux macOS **Windows**

Product/file description	File size	Download
x64 Compressed Archive	170.66 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 ↗)
x64 Installer	152 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 ↗)
x64 MSI Installer	150.89 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 ↗)

Pasos previos

[Learn about Java SE Subscription](#)

**Antes de instalar Android Studio debes tener instalado el JDK de Java.
Instala la última versión LTS que ahora mismo es el JDK 17:**

<https://www.oracle.com/es/java/technologies/javase-downloads.html>

Al lío... vamos a hacer nuestra primera App con Ionic...

Pasos previos

Instala Android Studio:

<https://developer.android.com/studio/install>



Android Studio



Al lío... vamos a hacer nuestra primera App con Ionic...

¿Qué tenemos ahora?

Tenemos instalado:

- **NodeJS**, que es lo que ha hecho que JavaScript pueda ejecutarse fuera del navegador web.
- **npm**, que es el gestor de paquetes de node. (Parecido a apt para Linux)
- Los comandos de abajo permiten ver la versión instalada.

```
$ node --version  
$ npm --version
```


Al lío... vamos a hacer nuestra primera App con Ionic...

Instalemos ionic...

Instalemos ionic:

- **@ionic/cli**, ionic viene como un paquete más disponible en npm.
- La opción **-g** es para instalarlo globalmente en nuestro equipo.

```
$ npm install -g @ionic/cli
```

Al lío... vamos a hacer nuestra primera App con Ionic...

Creemos
nuestra primera
app con ionic...

Creando nuestra primera app en ionic:

- **ionic start**, es para crear nuestro proyecto.
- **myApp**, es el nombre que le doy a mi proyecto.
- **blank**, es la plantilla de inicio. Otras opciones son tabs, sidemenu, etc...
- **--capacitor**, es que voy a usar integración con capacitor. La otra opción posible es --cordova
- **--type=angular**, es que voy a trabajar con angular. Otras opciones son react y vue. También puedes usar versiones anteriores: "ionic1" ó "ionic-angular". "ionic start --list" para un listado completo.

```
$ ionic start myApp blank --capacitor --type=angular
```


Al lío... vamos a hacer nuestra primera App con Ionic...

Creemos nuestra primera app con ionic...

Seguramente te preguntará lo siguiente si quieres crear una cuenta de Ionic...

Para esta práctica no necesitas una cuenta de Ionic...

Para ir al grano en esta práctica responde que No pulsando ENTER.

Join the Ionic Community! 

Connect with millions of developers on the Ionic Forum and get access to live events, news updates, and more.

? Create free Ionic account? (y/N)

Al lío... vamos a hacer nuestra primera App con Ionic...

Creemos nuestra primera app con ionic...

Si llegas a esta pantalla es que has conseguido crear el esqueleto de un proyecto con Ionic que ya está listo para trabajar...

Your Ionic app is ready! Follow these next steps:

- Go to your new project: `cd .\myApp`
- Run `ionic serve` within the app directory to see your app in the browser
- Run `ionic capacitor add` to add a native iOS or Android project using Capacitor
- Generate your app icon and splash screens using `cordova-res --skip-config --copy`
- Explore the Ionic docs for components, tutorials, and more: <https://ion.link/docs>
- Building an enterprise app? Ionic has Enterprise Support and Features: <https://ion.link/enterprise-edition>

```
tibur@MSI MINGW64 /c/MisCosas/Casa/Ionic
```

```
$ █
```

Al lío... vamos a hacer nuestra primera App con Ionic...

Ejecutamos nuestra primera App con Ionic...

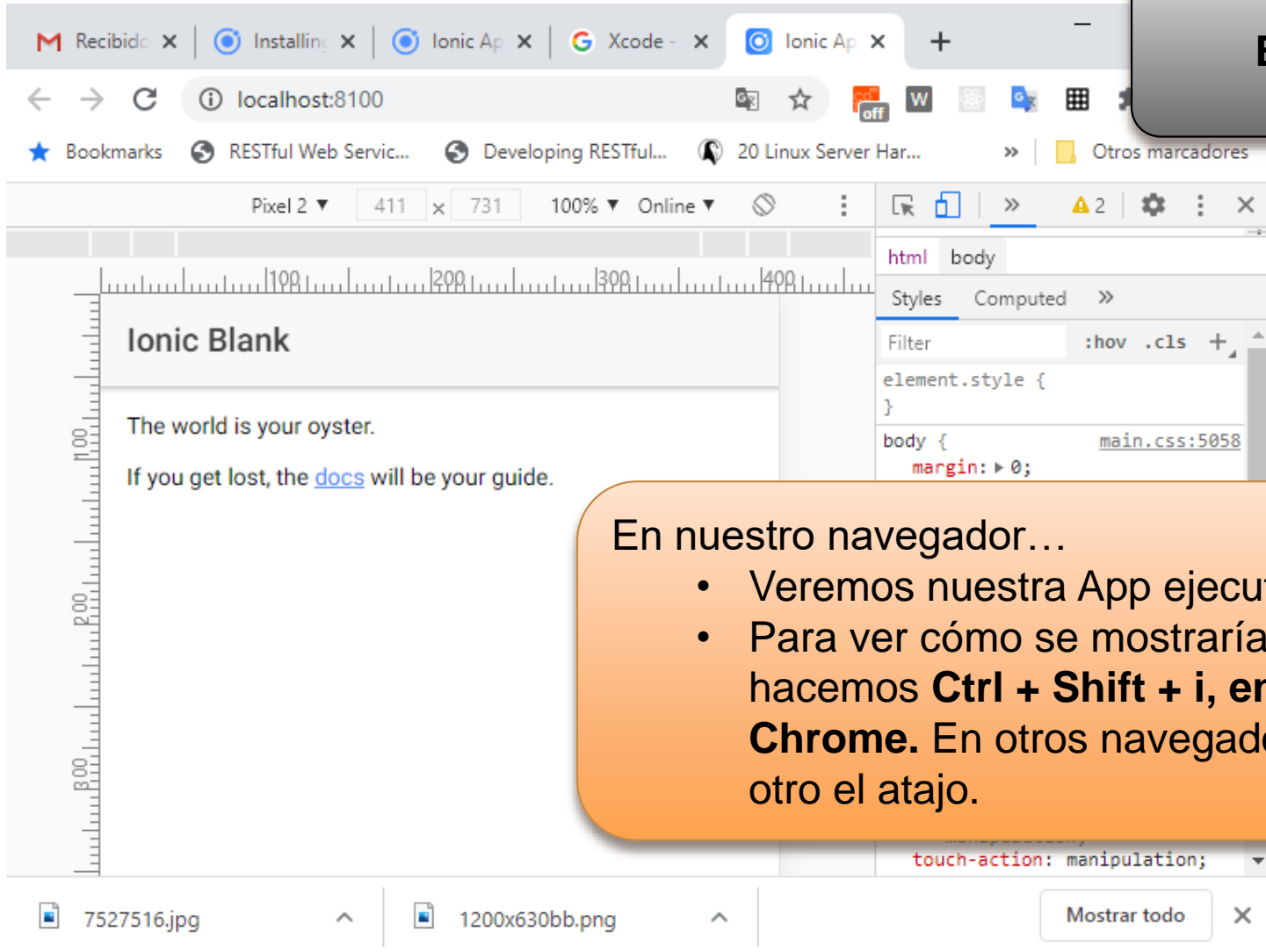
Ejecutamos nuestra primera App con Ionic:

- **cd myApp**, Cambiamos al directorio creado.
- **ionic serve**, Ejecutamos el simulador.

```
$ cd myApp  
$ ionic serve
```

Al lío... vamos a hacer nuestra primera App con Ionic...

Et voilà...



En nuestro navegador...

- Veremos nuestra App ejecutándose.
- Para ver cómo se mostraría en un móvil hacemos **Ctrl + Shift + i**, en **Google Chrome**. En otros navegadores puede ser otro el atajo.

Podemos usar el paquete @ionic/lab

@ionic/lab...

Resulta interesante ...

- Porque muestra el aspecto de la App como quedaría en cada plataforma.
- La opción **--save-dev** permite instalar la dependencia indicando que se va a usar en desarrollo. Es interesante que edites el fichero **package.json** para ver en la sección que se guarda.

```
$ npm install @ionic/lab --save-dev
```

```
$ ionic serve --lab
```

package.json

```
"dependencies": {
  "@angular/animations": "5.2.11",
  "@angular/common": "5.2.11",
  "@angular/compiler": "5.2.11",
  "@angular/compiler-cli": "5.2.11",
  "@angular/core": "5.2.11",
  "@angular/forms": "5.2.11",
  "@angular/platform-browser": "5.2.11",
  "@angular/platform-browser-dynamic": "5.2.11",
  "@ionic-native/core": "4.20.0",
  "@ionic-native/splash-screen": "4.20.0",
  "@ionic-native/status-bar": "4.20.0",
  "@ionic/storage": "2.2.0",
  "ionic-angular": "3.9.9",
  "ionicons": "3.0.0",
  "rxjs": "5.5.11",
  "sw-toolbox": "3.6.0",
  "zone.js": "0.8.29"
},
"devDependencies": {
  "@ionic/app-scripts": "3.2.4",
  "@ionic/lab": "^3.2.7",
  "typescript": "2.6.2"
},
```

Uso de package.json

Resulta interesante ...

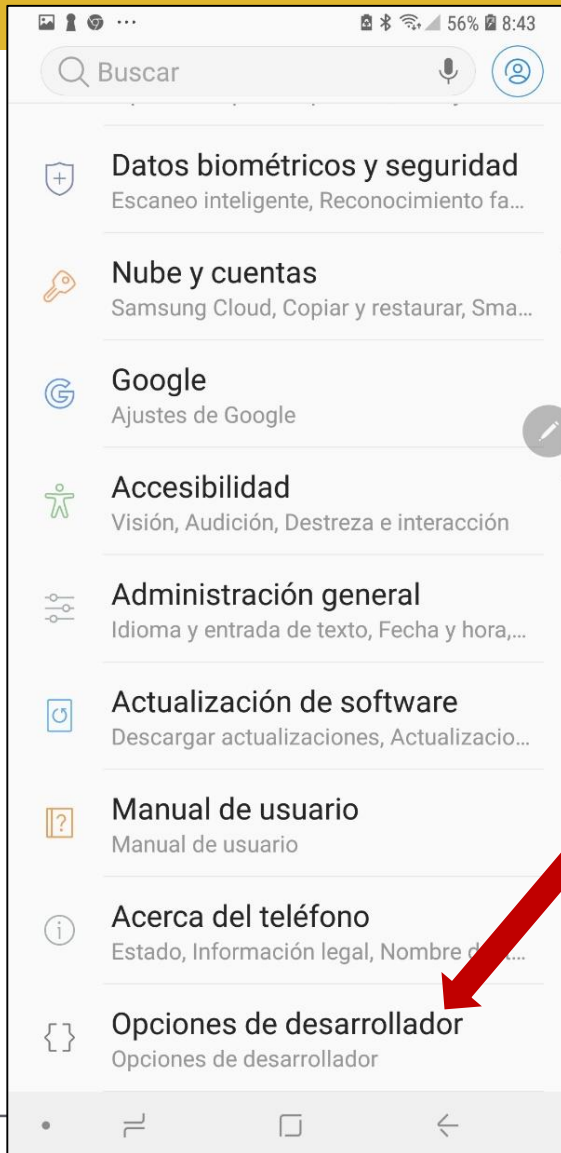
- En **dependencies** se encuentran los paquetes del proyecto.
- En **devDependencies** se encuentran los paquetes que has instalado con la opción **--save-dev** que son aquellos que se usan en desarrollo.

Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...

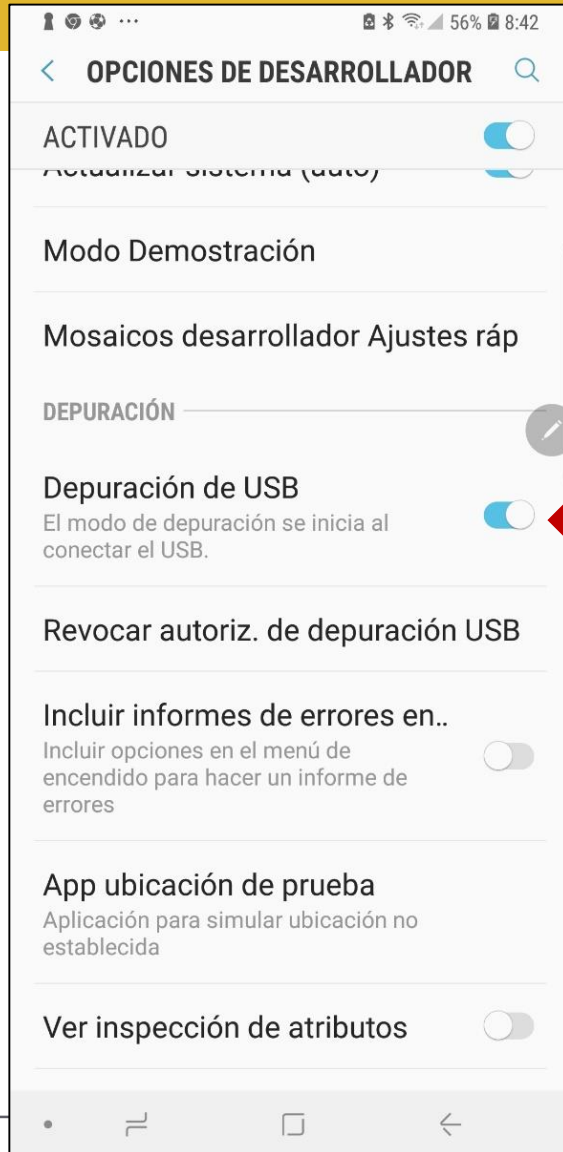
En tu móvil debes hacer visible la opción de **“Opciones de desarrollador”** oculta por defecto.

- En mi móvil está en Ajustes → Acerca del Teléfono → Información del Software → Número de compilación
- Sobre la opción **“Número de compilación”** debes pulsar 7 veces y entonces se verá la opción **“Opciones de desarrollador”**



Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...



Una vez visible la opción de
“Opciones de desarrollador”
accede a dicha opción y luego
habilita la **“depuración USB”**

Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...

(1) Conecta tu móvil
por USB a tu Equipo
de desarrollo

```
$ ionic build
```

```
$ ionic cap add android
```

```
$ ionic cap open android
```

(2) Con ello estas creando
tu proyecto www. Observa
que en tu proyecto aparece
dicha carpeta www.

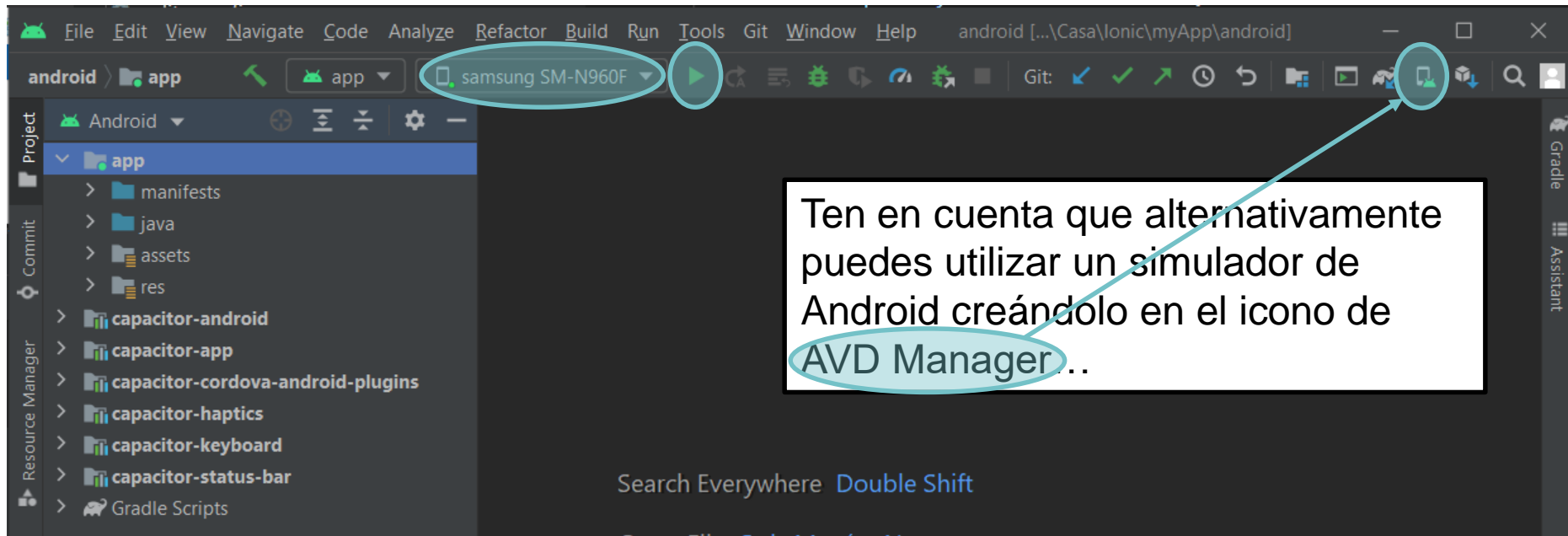
(3) Se crea la carpeta
"Android" con el
proyecto Android.

(4) Con este comando se abrirá
el Android Studio con tu proyecto

Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...

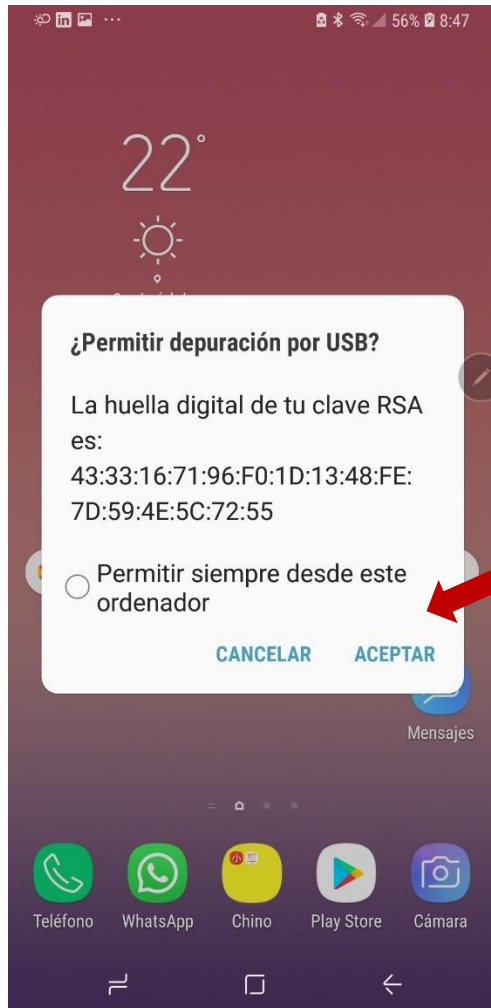
Ahora tu proyecto está de forma nativa abierto con el IDE Android Studio, y tu móvil debería aparecer (en mi caso un Samsung SM-N960F) y sólo tendrías que hacer clic en el icono verde de play para ejecutar tu proyecto en tu móvil.



Ten en cuenta que alternatively
puedes utilizar un simulador de
Android creándolo en el icono de
AVD Manager...

Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...



Al aceptar ya se ejecutará
tu App en tu móvil.

... y si después haces cambios en el código fuente de tu proyecto...

Prueba tu App en tu móvil...

Si quieres verlos en el móvil entonces debes ejecutar los siguientes comandos...

```
$ ionic build
```

```
$ ionic cap copy
```

```
$ ionic cap sync
```

(2) Con ello estas actualizando el contenido de tu proyecto en la carpeta www.

(3) Se copian los cambios a las diferentes plataformas. En este caso a Android.

(4) Con este comando se sincronizarán los proyectos de las diferentes plataformas. En nuestro caso para que lo puedas ver desde el Android Studio.

Sigue aprendiendo...

Sigue el siguiente ejemplo paso a paso que se explica en la web oficial de Ionic:

- <https://ionicframework.com/docs/angular/your-first-app>

¡Cuidado con las versiones!

A fecha de la creación de este tutorial deberías trabajar con la versión 5 de Ionic.

Los tutoriales de Ionic 4 te pueden aún servir en su mayoría.

Pero los anteriores ya tienen bastantes diferencias.

Conclusiones

¿Qué hemos aprendido?

- Simplemente hemos instalado el entorno, creando una App y la hemos probado en un móvil.
- También hemos visto las posibilidades más usuales para crear apps para móvil.

Próximos pasos...

- Hemos usado Angular como framework. Así que lo próximo es empezar a trabajar con Angular en Ionic.
-