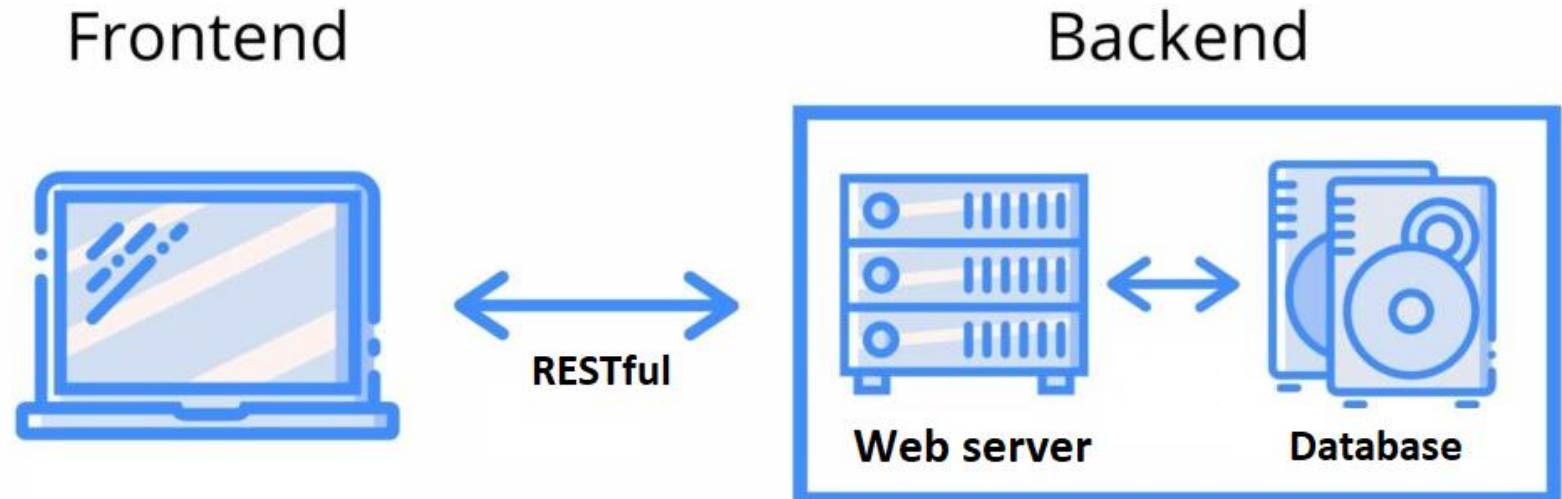


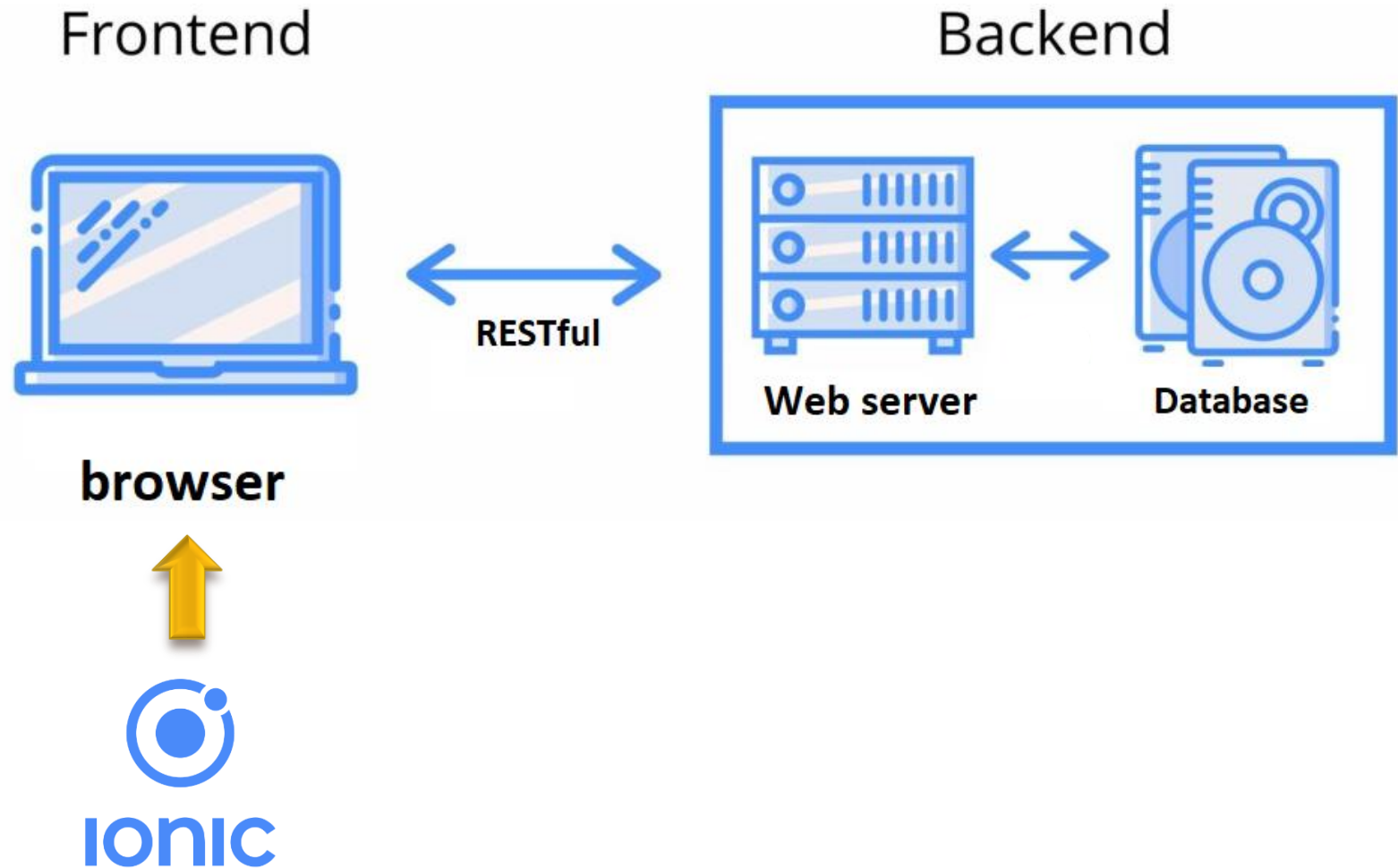
Introducción a IONIC (Usando Angular)

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

¿Dónde corre Ionic?



¿Dónde corre Ionic?

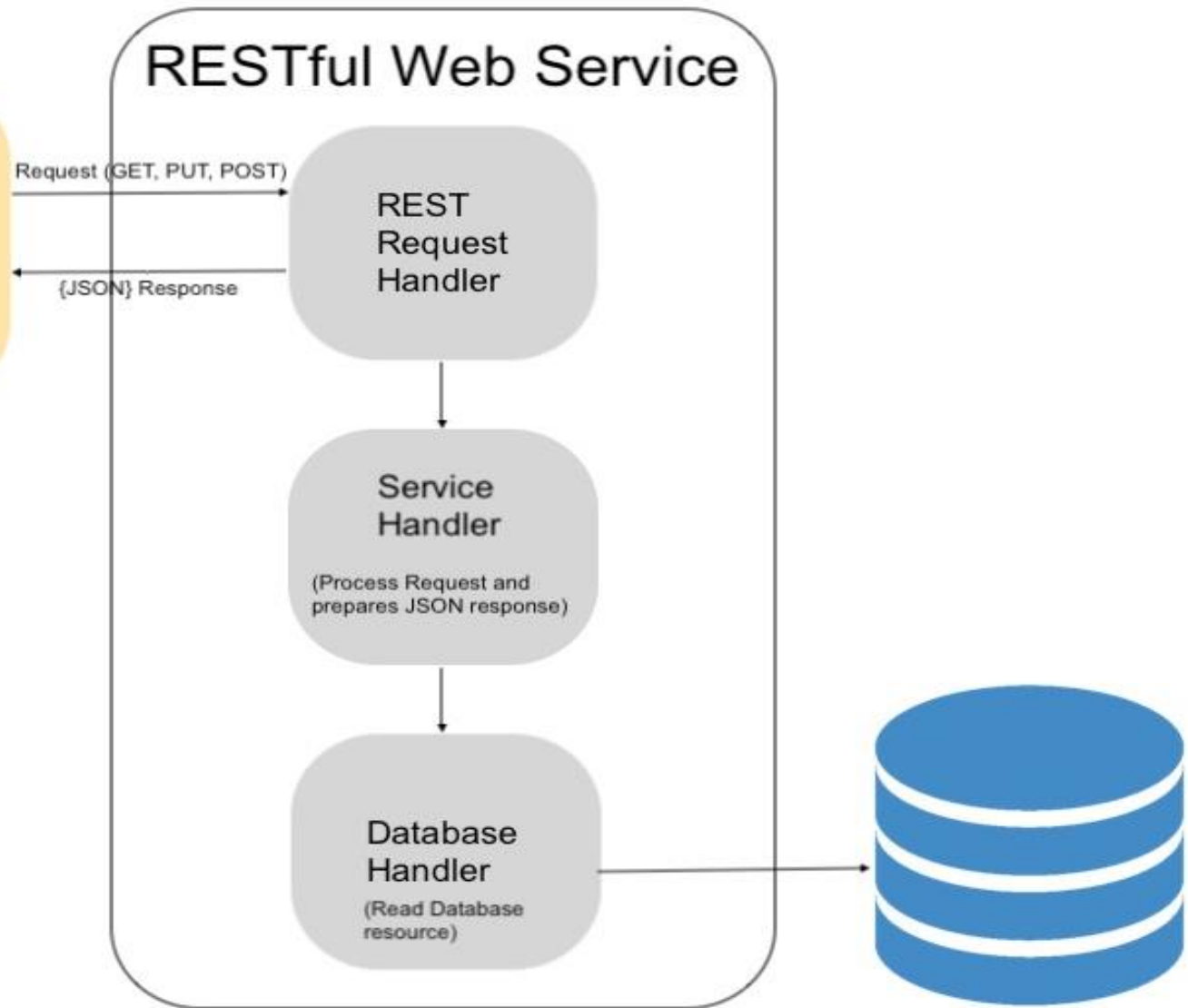


No perdamos nunca la visión global que perseguimos...

IONIC

REST Client

Un móvil, aplicación de escritorio ó navegador web



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/download>

Obtiene Node.js® v22.19.0 (LTS) para Windows usando Docker con npm



Windows Installer (.msi)



Standalone Binary (.zip)

Descarga e instala la
versión LTS de NodeJS

nodejs.org/es/download

Obtiene Node.js® v22.19.0 (LTS) para Windows usando Docker con npm

Info Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!

```
1 # Docker provee instrucciones dedicadas para cada sistema operativo
2 # Por favor consulta la documentación oficial en https://docs.docker.com/
3
4 # Descarga la imagen de Docker de Node.js:
5 docker pull node:22-alpine
6
7 # Crea un contenedor de Node.js e inicia una sesión:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.19.0".
12
13 # Verifica versión de npm:
14 npm -v # Debería mostrar "10.9.3".
```

PowerShell

Copiar al portapapeles

Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)

O obtiene una versión pre compilada de Node.js® para Windows usando la arquitectura

x64



Windows Installer (.msi)



Standalone Binary (.zip)

La página completa por si
no lo viste bien en la
transparencia anterior

Descarga e instala la
versión LTS de NodeJS

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

Pasos previos

JDK 21 is the latest *Long-Term Support (LTS)* release of the Java SE Platform.

Earlier JDK versions are available below.

JDK 24 **JDK 21** GraalVM for JDK 24 GraalVM for JDK 21

Última LTS siempre

Java SE Development Kit 21.0.8 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#) (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License](#) (OTN) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

Linux macOS **Windows**

Product/file description	File size	Download
x64 Compressed Archive	186.05 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	164.42 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)

**Antes de instalar Android Studio debes tener instalado el JDK de Java.
Instala la última versión LTS que ahora mismo es el JDK 21:**

<https://www.oracle.com/es/java/technologies/downloads/#jdk21-windows>

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

Pasos previos

Instala Android Studio:

<https://developer.android.com/studio?hl=es-419>



Android Studio



¡NO ES SUPER NECESARIO!

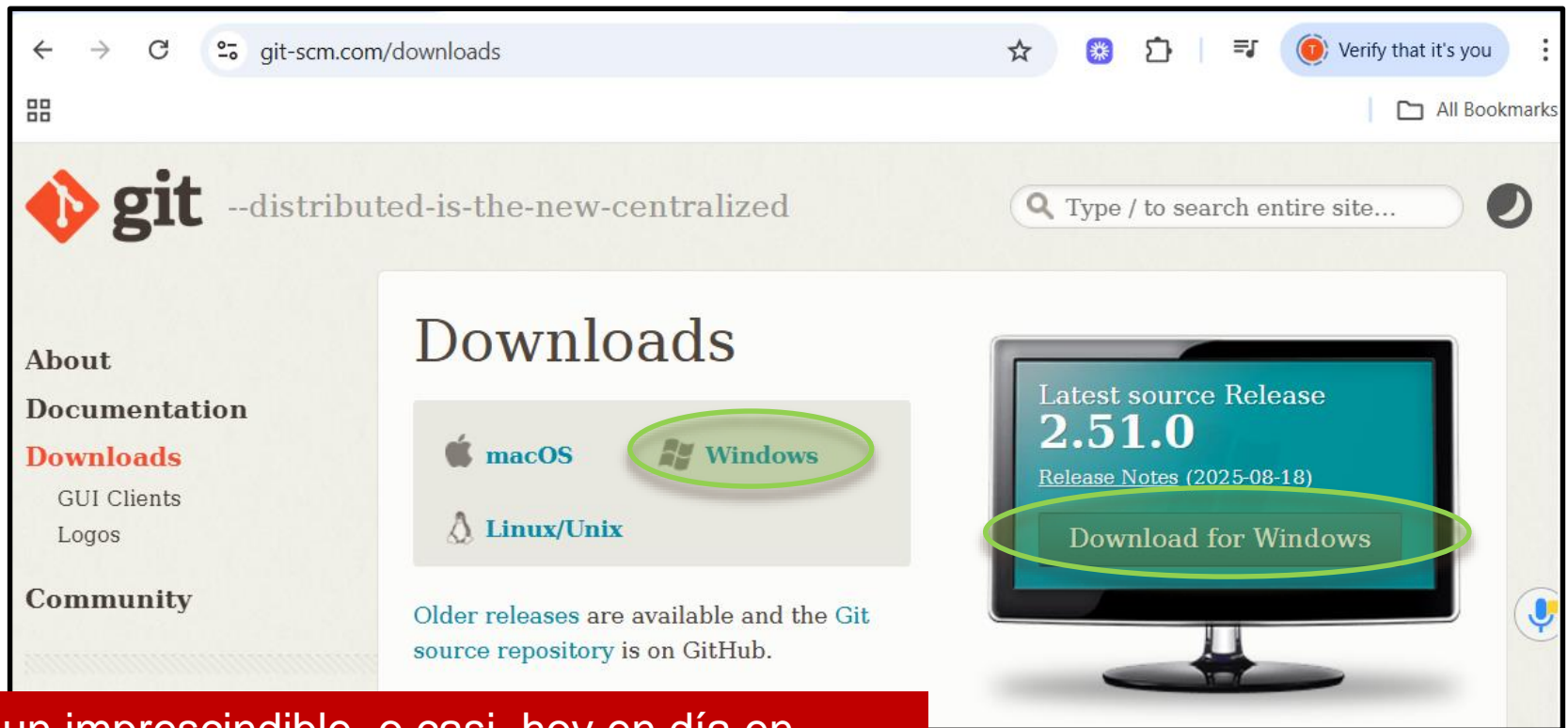
Sólo si quieres crear tu .apk para ejecutarlo en tu móvil

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

Instala Git:

<https://git-scm.com/downloads>

Pasos previos



Es un imprescindible, o casi, hoy en día en desarrollo. El gestor de versiones más usado.

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
```

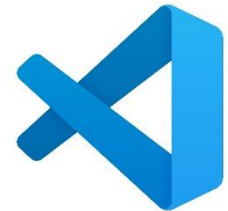
Al lío... vamos a hacer nuestra primera App con Ionic...

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.

...pero antes... instala **visual studio code**. Es uno de los editores más usados hoy en día...



```
$ 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
```

***Nota:** elige NgModules cuando te salga la opción*


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

Ejecutamos nuestra primera App con Ionic...

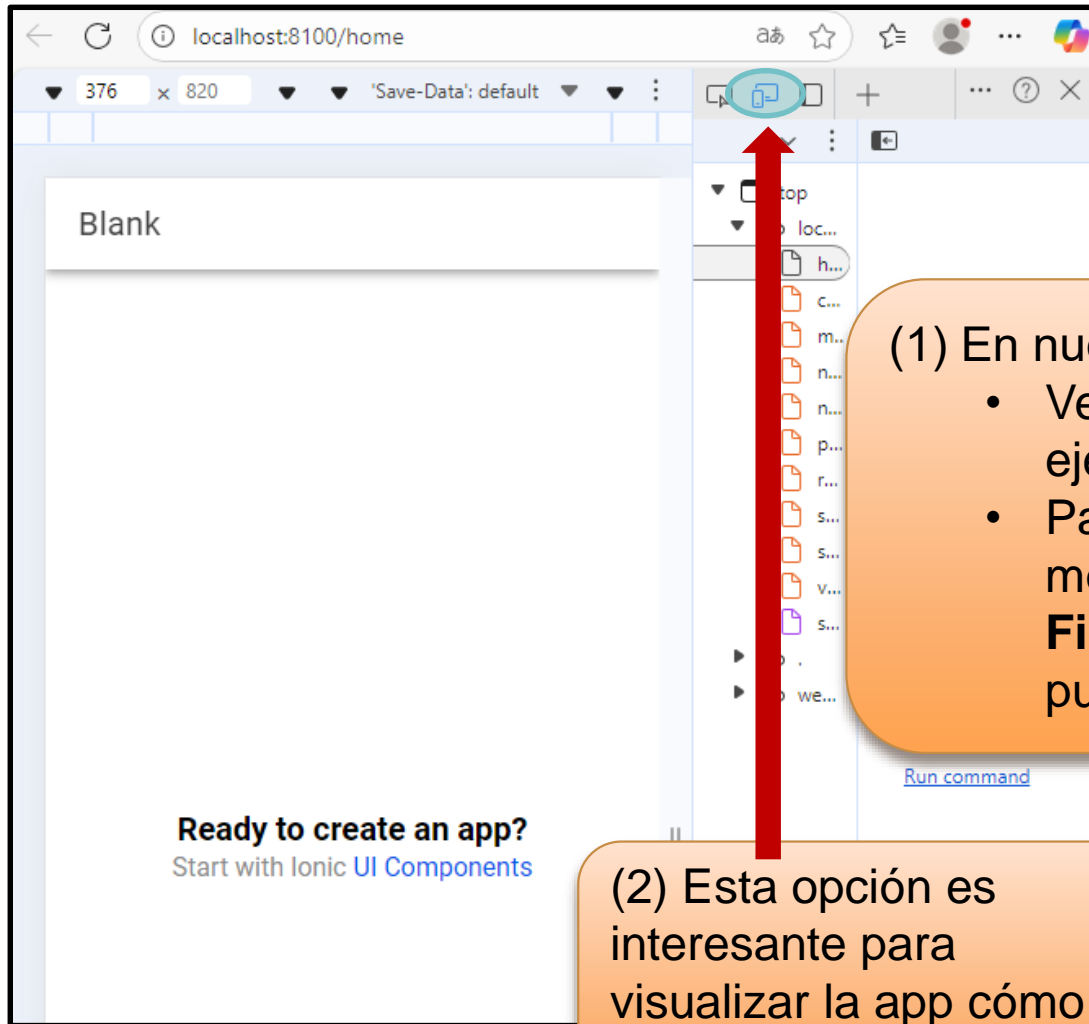
Ejecutamos nuestra primera App con Ionic:

- **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...



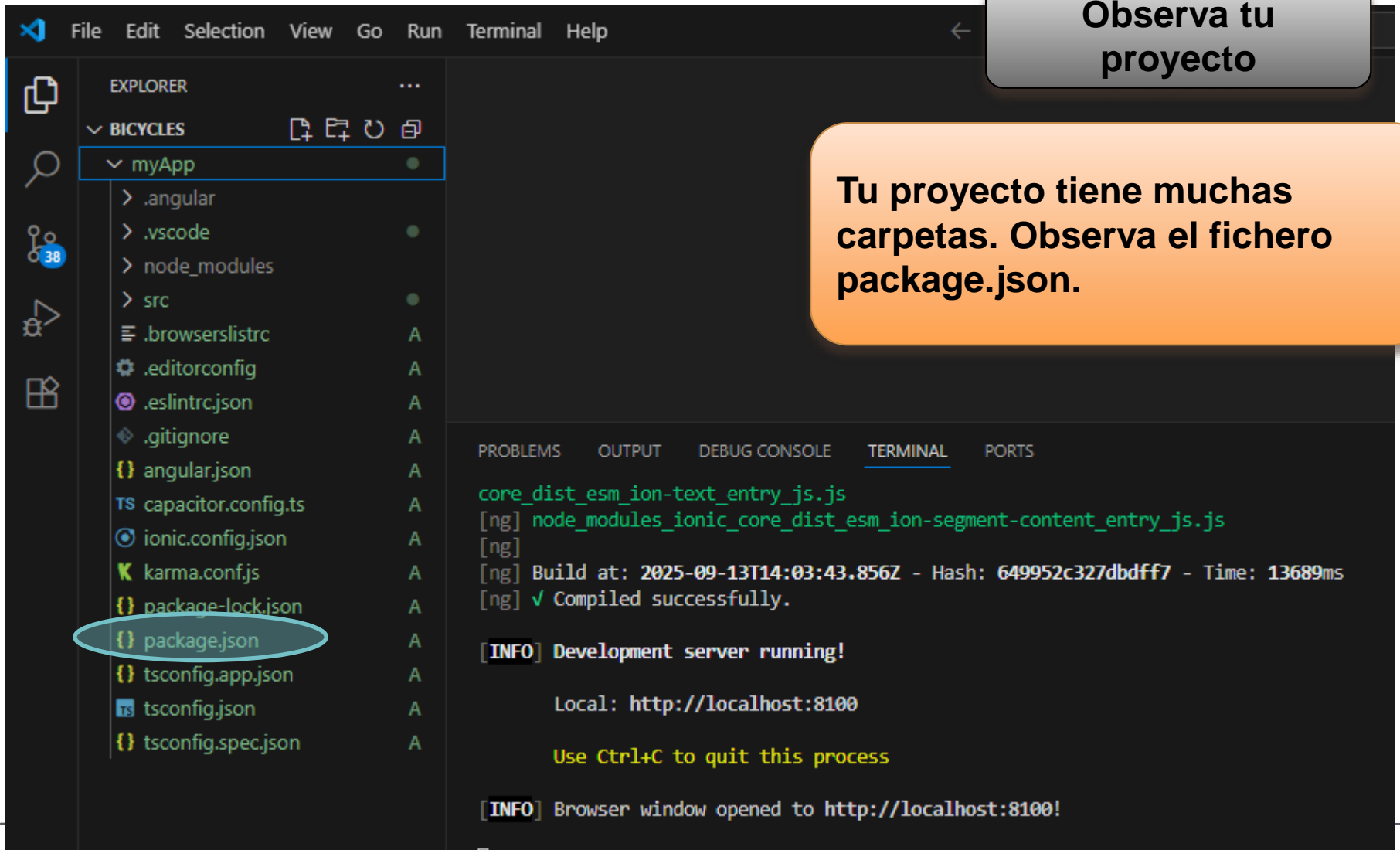
- (1) En nuestro navegador...
- Veremos nuestra App ejecutándose.
 - Para ver cómo se mostraría en un móvil hacemos **Ctrl + Shift + i**, en **Firefox**. En otros navegadores puede ser otro el atajo.

(2) Esta opción es interesante para visualizar la app cómo se vería en un móvil.

Tu proyecto en el Visual Studio Code

Observa tu proyecto

Tu proyecto tiene muchas carpetas. Observa el fichero package.json.



package.json

Uso de package.json

```
"dependencies": {  
  "@angular/animations": "^20.0.0",  
  "@angular/common": "^20.0.0",  
  "@angular/compiler": "^20.0.0",  
  "@angular/core": "^20.0.0",  
  "@angular/forms": "^20.0.0",  
  "@angular/platform-browser": "^20.0.0",  
  "@angular/platform-browser-dynamic": "^20.0.0",  
  "@angular/router": "^20.0.0",  
  "@capacitor/app": "7.1.0",  
  "@capacitor/core": "7.4.3",  
  "@capacitor/haptics": "7.0.2",  
  "@capacitor/keyboard": "7.0.3",  
  "@capacitor/status-bar": "7.0.3",  
  "@ionic/angular": "^8.0.0",  
  "ionicons": "^7.0.0",  
  "rxjs": "~7.8.0",  
  "tslib": "^2.3.0",  
  "zone.js": "~0.15.0"  
},  
"devDependencies": {  
  "@angular-devkit/build-angular": "^20.0.0",  
  "@angular-eslint/builder": "^20.0.0",  
  "@angular-eslint/eslint-plugin": "^20.0.0",  
  "@angular-eslint/eslint-plugin-template": "^20.0.0",  
  "@angular-eslint/schematics": "^20.0.0",  
  "@angular-eslint/template-parser": "^20.0.0",  
  "@angular/cli": "^20.0.0",
```

package.json contiene las dependencias del proyecto.

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.

A partir de aquí veremos
como ejecutar tu App en un
móvil Android o en un
Simulador de móvil Android.

**Necesitas tener instalado
Android Studio**

Prueba tu App en tu móvil o en un simulador...

- (1) Conecta tu móvil por USB a tu Equipo de desarrollo
- (2) Si tienes ejecutando tu proyecto pulsa Ctrl-C para pararlo.

Prueba tu App en tu móvil...

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

```
$ ionic build
```

```
$ ionic cap add android
```

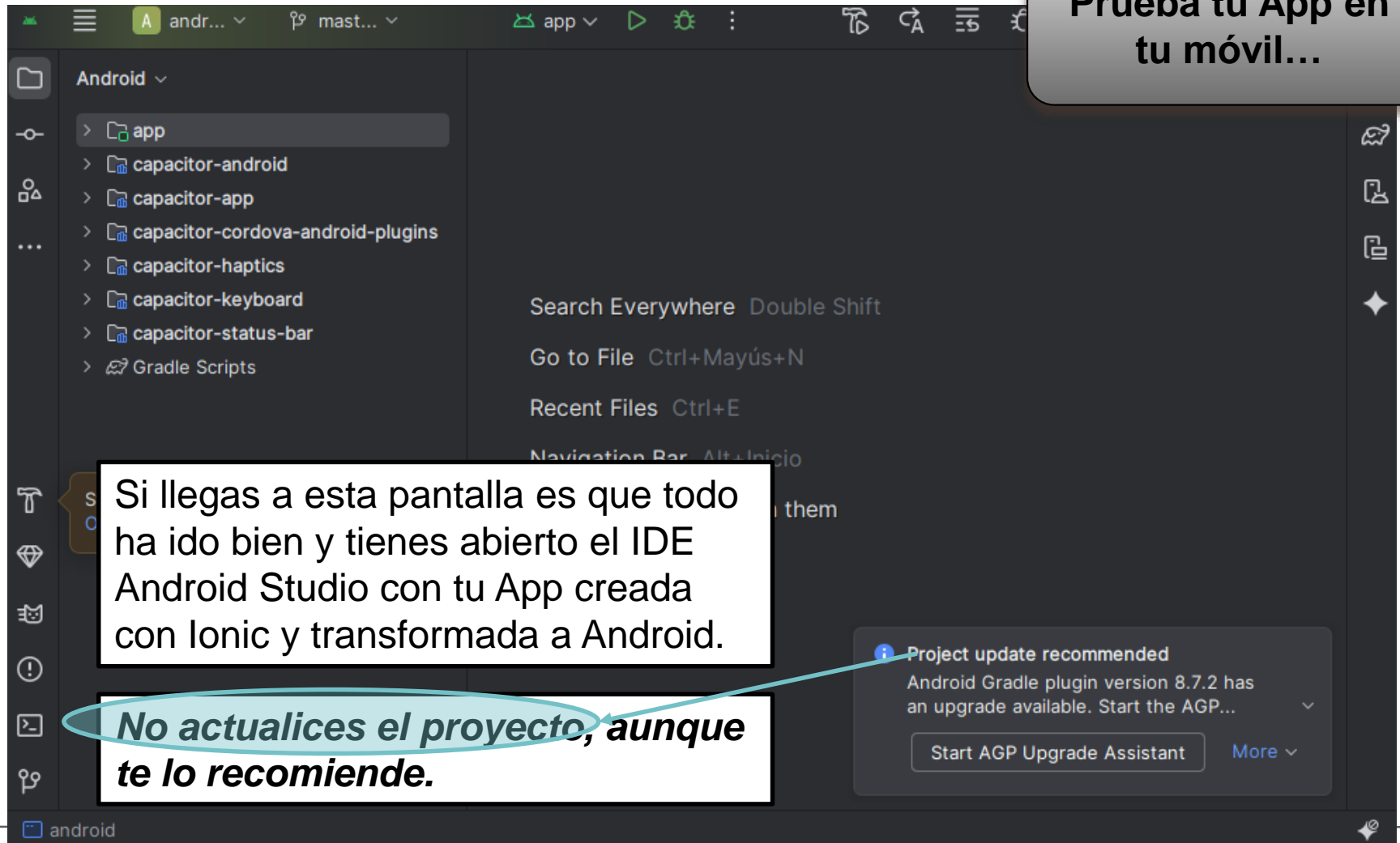
```
$ ionic cap open android
```

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

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

Prueba tu App en tu móvil o en un simulador...

Prueba tu App en
tu móvil...



Si llegas a esta pantalla es que todo ha ido bien y tienes abierto el IDE Android Studio con tu App creada con Ionic y transformada a Android.

No actualices el proyecto, aunque te lo recomienda.

Project update recommended
Android Gradle plugin version 8.7.2 has an upgrade available. Start the AGP...
[Start AGP Upgrade Assistant](#) [More](#)

Prueba tu App en un simulador...

Prueba tu App en
tu móvil...

*Si le das al Play ejecutarás tu App
en un simulador por defecto.*

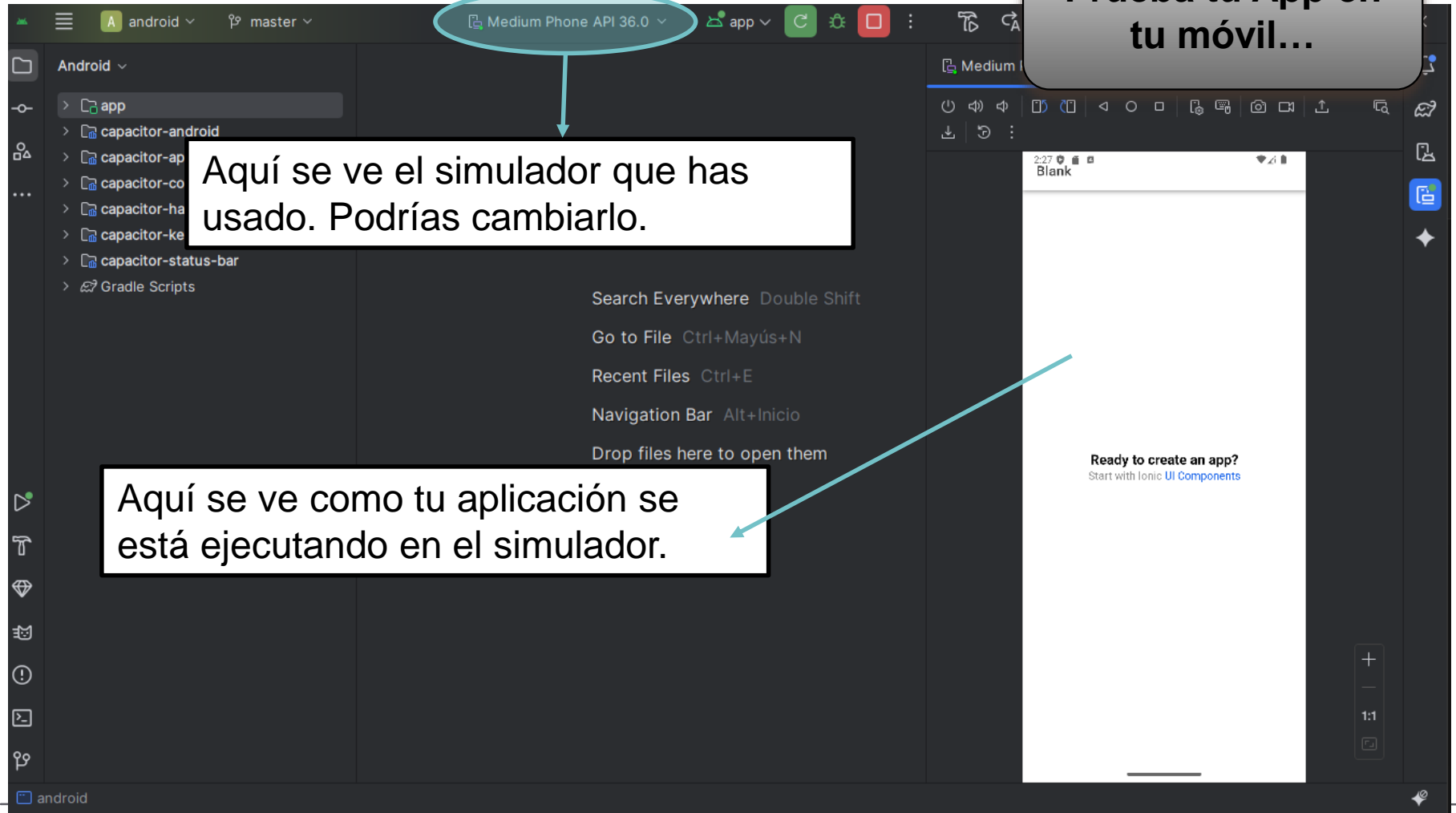
Si llegas a esta pantalla es que todo
ha ido bien y tienes abierto el IDE
Android Studio con tu App creada
con Ionic y transformada a Android.

**No actualices el proyecto, aunque
te lo recomiende.**

Project update recommended
Android Gradle plugin version 8.7.2 has
an upgrade available. Start the AGP...
[Start AGP Upgrade Assistant](#) [More](#)

Prueba tu App en un simulador...

Prueba tu App en
tu móvil...

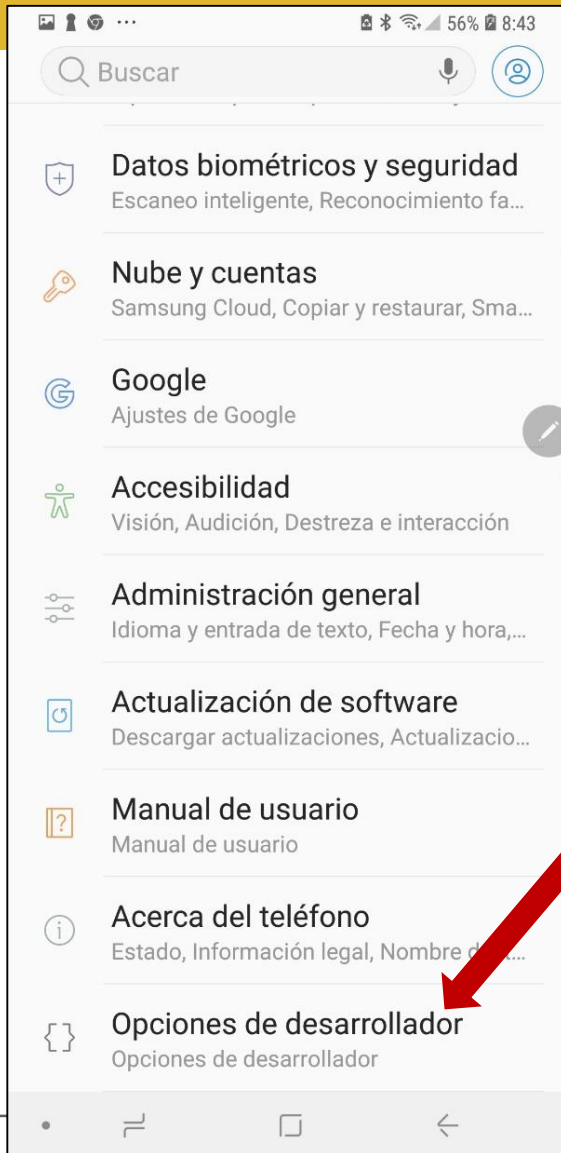


Prueba tu App en tu móvil...

Prueba tu App en
tu móvil...

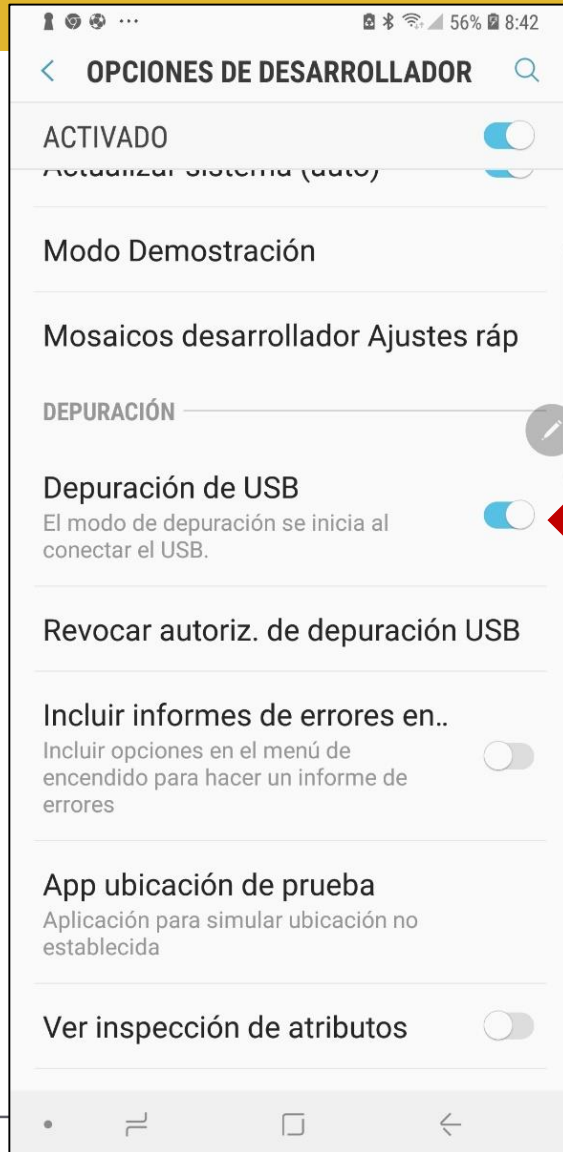
En tu móvil debes hacer visible la opción de **“Opciones de desarrollador”** oculta por defecto.

- 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...

Ahora tu proyecto está de forma nativa abierto con el IDE Android Studio, y tu móvil debería aparecer (en mi caso un Samsung SM-S918B) y sólo tendrías que hacer clic en el icono verde de play para ejecutar tu proyecto en tu móvil.

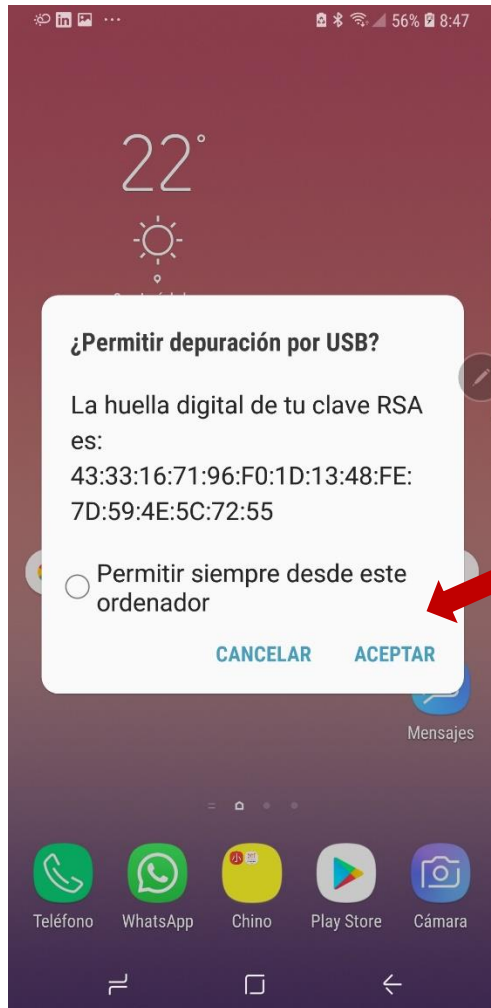
Prueba tu App en tu móvil...

Ten en cuenta que alternatively puedes utilizar un simulador de Android...

A veces es posible que tengas problemas para detectar tu móvil. La opción de "troubleshoot..." te puede ayudar.

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 8 de Ionic.

Los tutoriales incluso de la versión de Ionic 5 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.
-