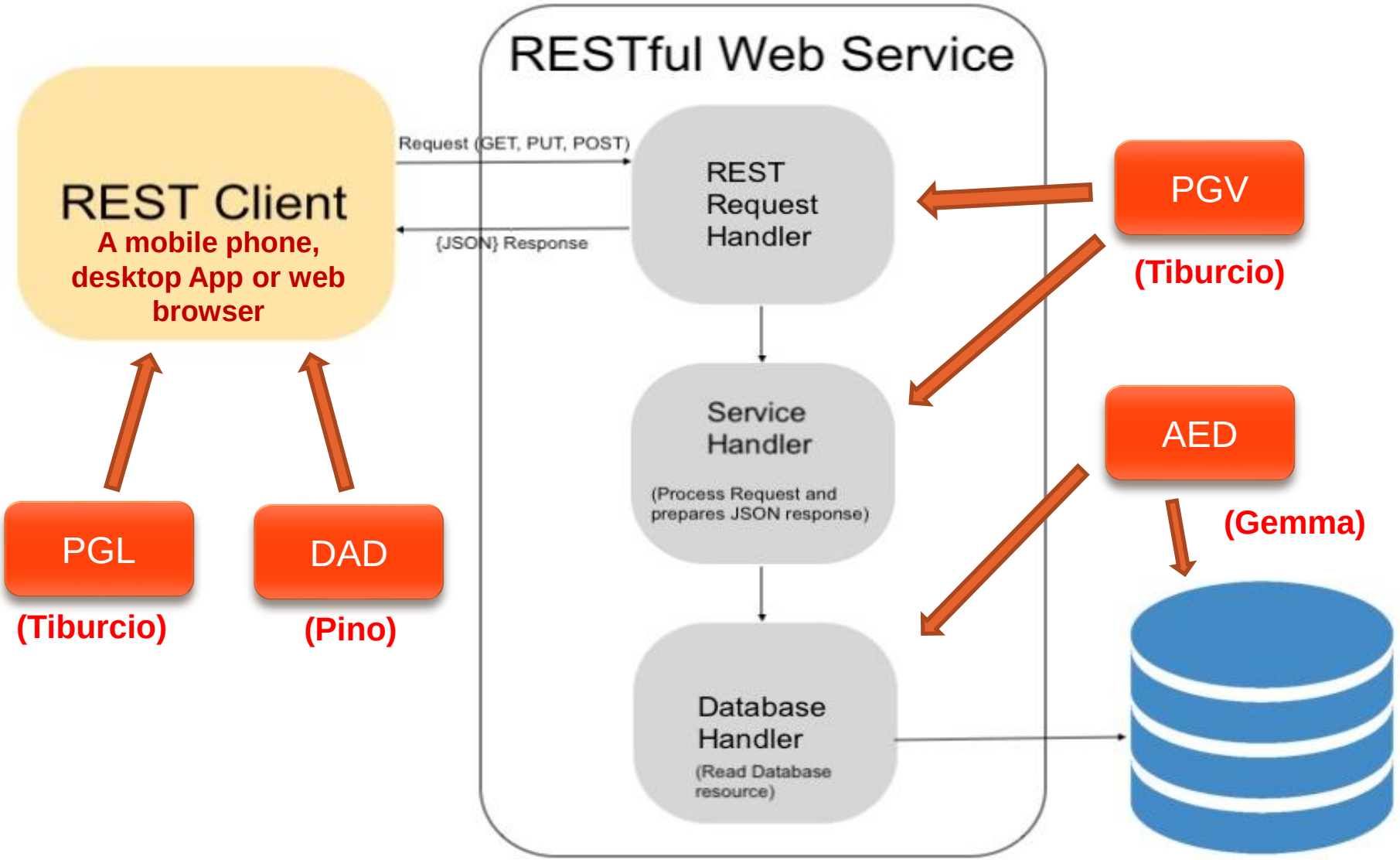
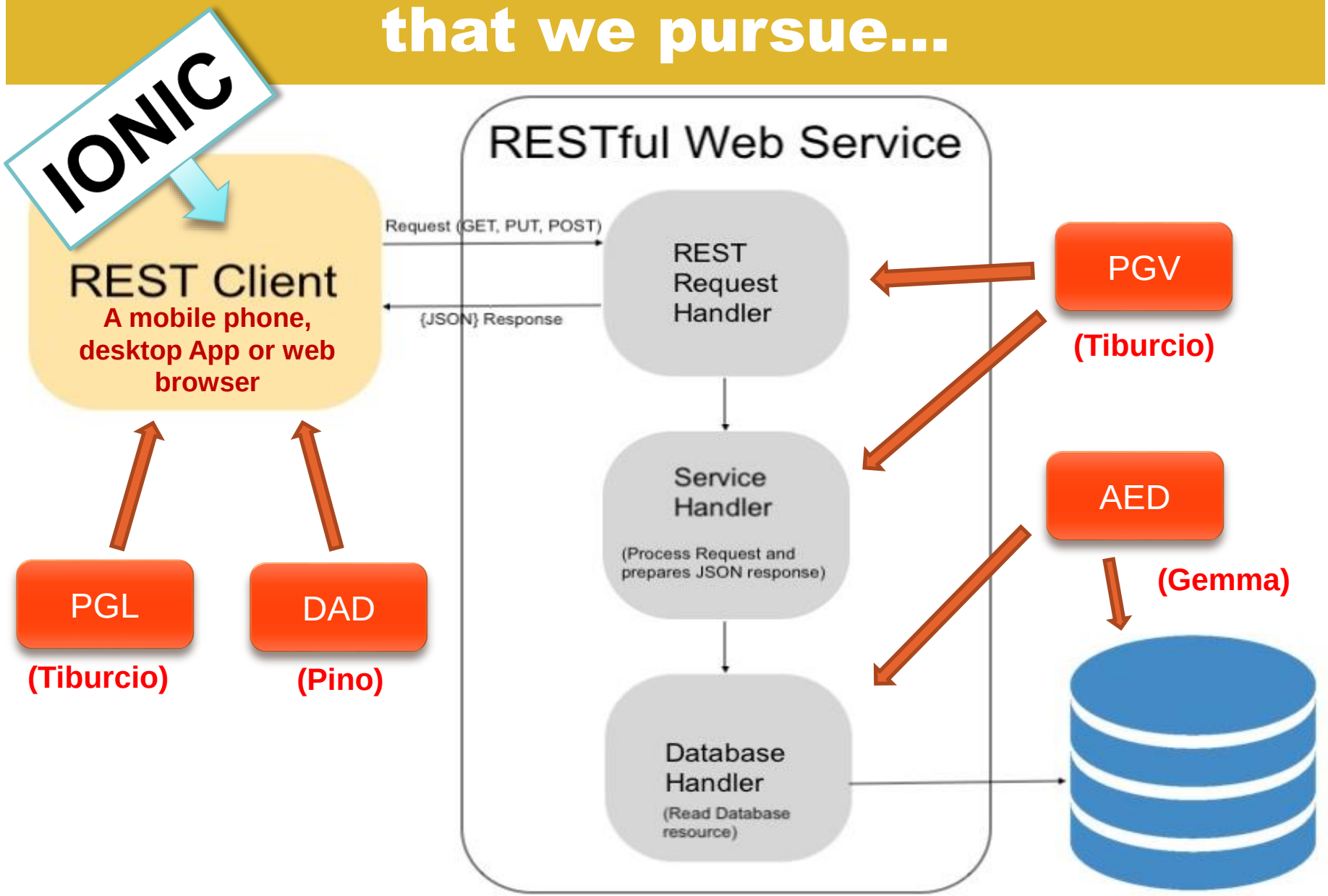
Introduction to IONIC (Using Angular)

Summary of steps based on the official Ionic page:
https://ionicframework.com/docs

Let us never lose the global vision that we pursue...



Let us never lose the global vision that we pursue...



Hybrid vs Native

Here is a good link to understand the difference: https://go.ionicframework.com/hybrid-vs-native-guide

(download the book)

At first you could only use the set of tools specific to each device:

- Android, programming in Java (or now in Kotlin) with Android Studio IDE.
- iOS, programming in Swift with Xcode IDE.

Hybrid vs Native

The main disadvantage of native apps is that you have to program from 0, so that your project works on each of the platforms.

With native apps you have to program the same project at least 3 times

Hybrid vs Native

Hybrid Apps follow the principle: “**code once, run anywhere**”

- You code with JS, HTML and CSS.
- The code runs in a web view (a browser takes up the entire screen)
- This web view is embed in a native wrapper which is different in each platform.
- This native wrapper is indeed the one which runs in the actual device.

Hybrid vs Native

The main disadvantage of native apps is that you have to program from 0, so that your project works on each of the platforms.

Hybrid apps however:

- They follow the principle: "**code once, run anywhere**".
- In Ionic you code with **HTML, CSS** and **JavaScript**.
- Then using **cordova** or **capacitor** you can create a version for **Android**, **iOS** or the **Web**.
- The code actually runs in a webview (full window size browser) that is embedded in an Android or iOS "native wrapper" that allows the App to behave like a native app.
- Access to the native resources of the device is done through plugins (for example for the camera).

What else is out there?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)

- They use the development tools (SDK) of each platform. Today Android and iOS.
- It is distributed through repositories such as Google play (Android), or App Store (iOS).
- You have to install the App to use it.

What else is out there?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)

- You code with JS, HTML and CSS.
- The code runs in a web view (a browser that occupies the entire screen)
- This webview is embedded in a native wrapper for each device.
- That native wrapper is actually the one running on the actual device.
- Hybrid Apps can also generate a web app with the same code.

What else is out there?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)

- It runs in a browser.
- Therefore it works on any device with a browser and the Internet.
- If there is no coverage it does not work.
- There is nothing to install except the browser.

What else is out there?

Native apps

Hybrid apps

Web apps

PWAs
(Progressive Web Apps)

- It is an intermediate solution between a web app and a native app.
- That is to say: it has an interface similar to that of a Native App. They work without an internet connection. Allows the sending of Push Notifications. It is updated automatically. It is Installable. Can be shared via URL.

What is Ionic?

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

Basically:

- It's **open source**.
- It's a set of tools (**toolkit**).
- Allows you to create cross-platform app experiences both native (android and ios), as well as web.
- You code **HTML, CSS and JavaScript**.
- It's "**framework agnostic**". You can use **React, Angular, Vue** or even just JavaScript (**JavaScript Vanilla**)
- **Code once, run anywhere**.

To the mess... we are going to make our first App with Ionic...

Previous steps

In the official Ionic page: https://nodejs.org/es/

Download and install the LTS NodeJS version

To the mess... we are going to make our first App with Ionic...

Previous steps

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.

Learn about Java SE Subscription

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)

Before installing Android Studio you must have the Java JDK installed. Install the latest LTS version which is JDK 11 right now :
https://www.oracle.com/es/java/technologies/javase-downloads.html

To the mess... we are going to make our first App with Ionic...

Previous steps

Install Android Studio:
https://developer.android.com/studio/install

To the mess... we are going to make our first App with Ionic...

And now what?

We have installed:

- **NodeJS**, that is what has allowed JavaScript to run outside of the web browser.
- **npm**, which is node's package manager. (similar to **apt** in Linux)
- The commands below allow you to see the installed version.

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

To the mess… we are going to make our first App with Ionic…

Let us install ionic …

Let us install ionic:

- **@ionic/cli**, ionic comes as a package available in npm.
- Option **–g** lets install a package globally in the computer.

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

To the mess… we are going to make our first App with Ionic…

Let us create our first ionic app…

Creating our first app in ionic:

- **ionic start,** is to create our project.
- **myApp,** is the name that I give to my project.
- **blank**, is the start template. Other options are: tabs, sidemenu, etc…
- **--capacitor,** is that I'm going to use capacitor integration. The other possible option is --cordova
- **--type=ionic-angular**, is that I am going to work with angular. Other options are react and vue. You can also use other old versions like “ionic1” or “ionic-angular”. “ionic start --list” shows all possible versions.

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

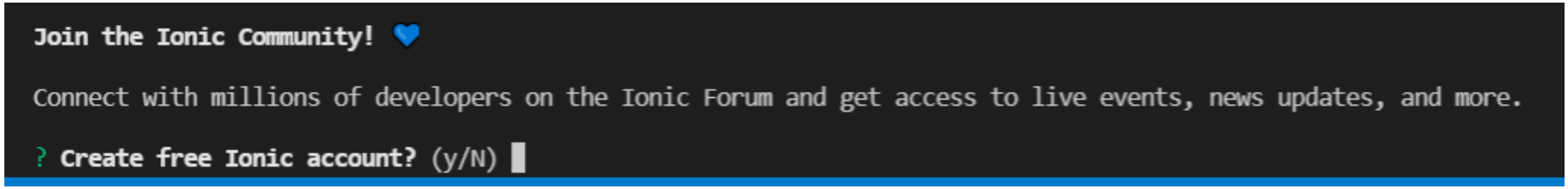
To the mess… we are going to make our first App with Ionic…

Let us install ionic …

It will surely ask you if you want to create an Ionic account...

You do not need an Ionic account for this exercise…

To get to the point in this exercise, answer No by pressing ENTER.



To the mess… we are going to make our first App with Ionic…

Let us install ionic …

If you get to this screen, you have managed to create the skeleton of a project with Ionic that is now ready to work…

```
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
$
```

To the mess… we are going to make our first App with Ionic…

Let us run our first ionic App…

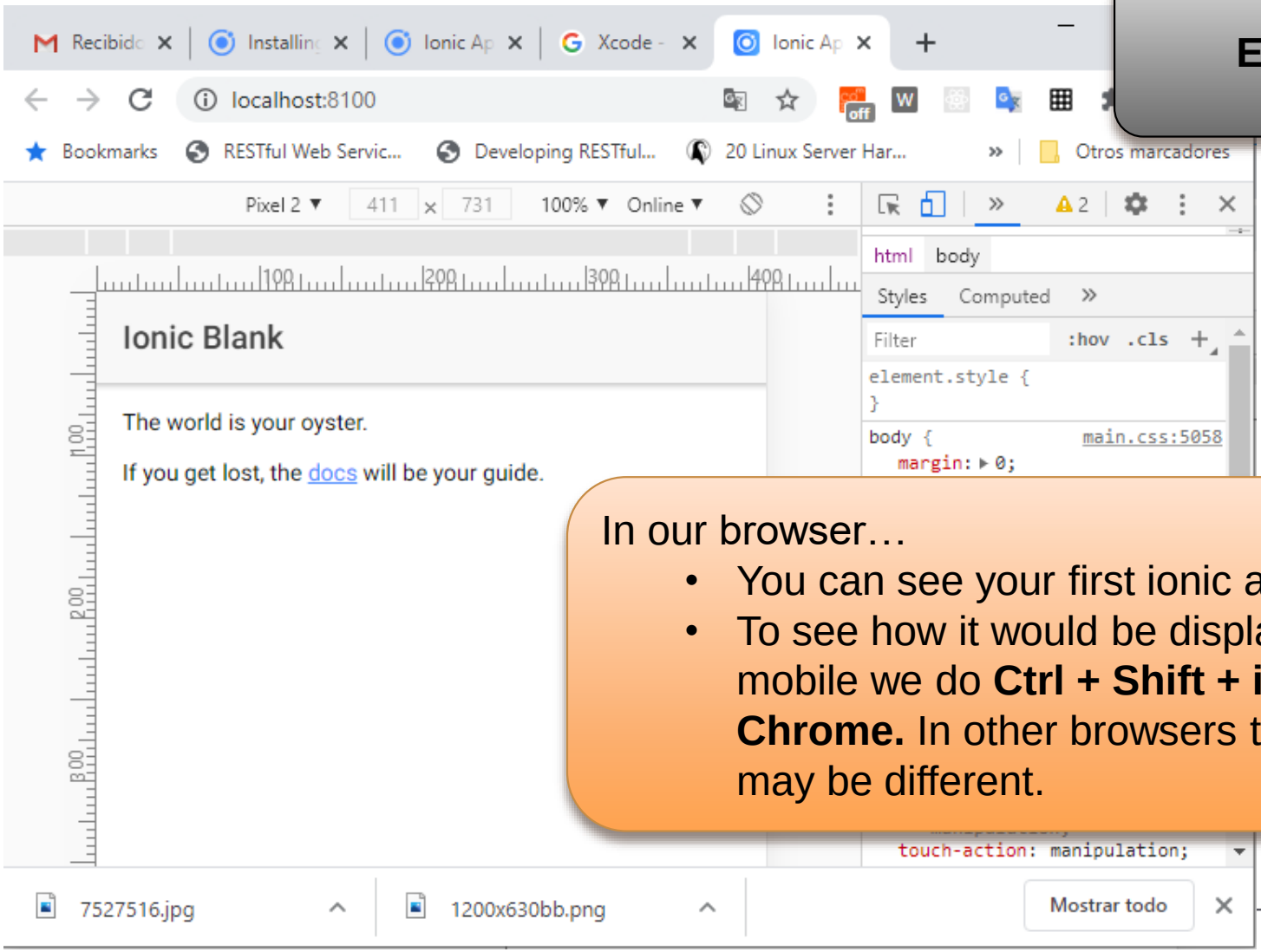
Let us run our first ionic App:

- **cd myApp,** to change to the created directory.
- **Ionic serve,** let us run the simulator.

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

To the mess… we are going to make our first App with Ionic…

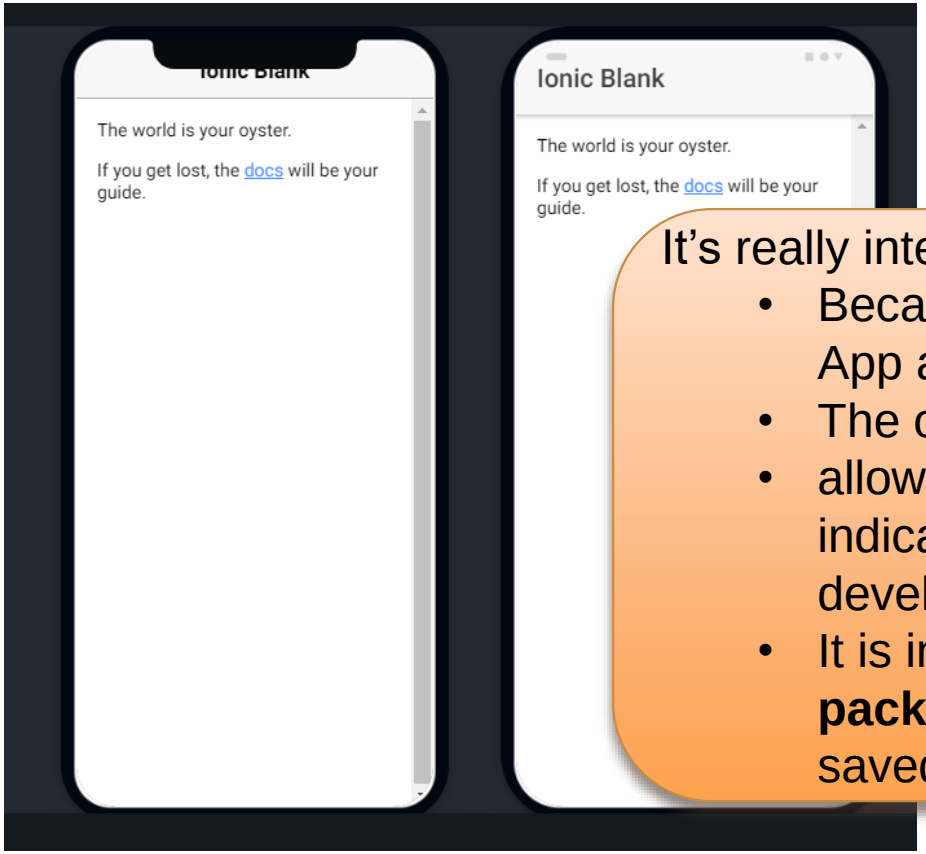
Et voilá…



In our browser…

- You can see your first ionic app running.
- To see how it would be displayed on a mobile we do **Ctrl + Shift + i,** in **Google Chrome.** In other browsers the shortcut may be different.

We can use the package @ionic/lab



@ionic/lab…

It's really interesting…

- Because it shows the appearance of the App as it would look on each platform.
- The option **--save-dev**
- allows you to install the dependency indicating that it will be used in development.
- It is interesting that you edit the file **package.json** to see in the section that is saved.

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

package.json

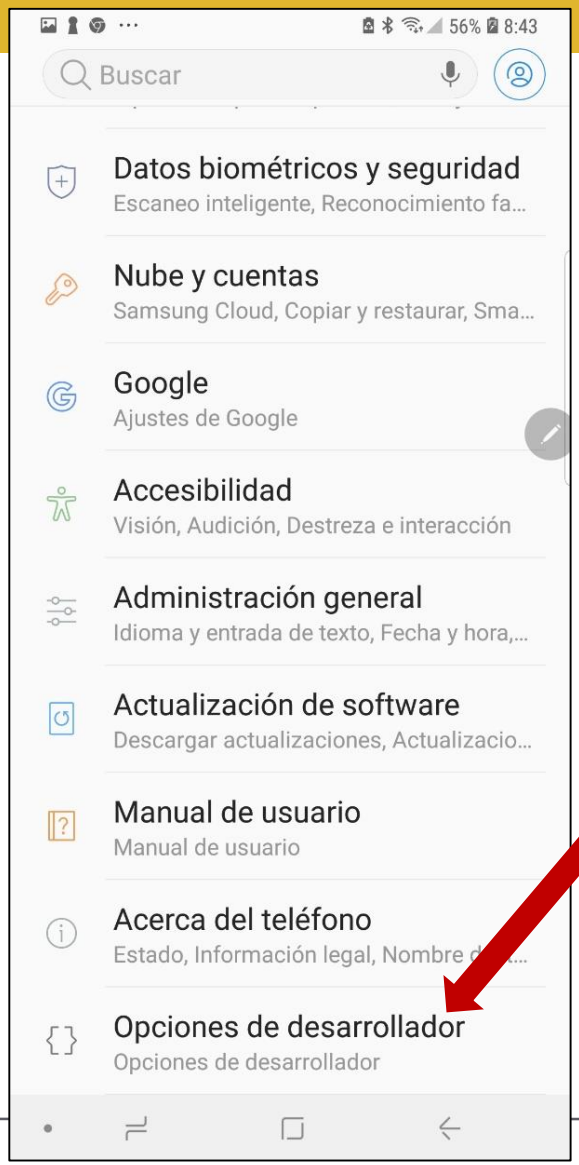
```
"dependencies": {
  "@angular/animations": "5.2.11",
  "@angular/common": "5.2.11",
  "@angular/compiler": "5.2.11",
  "@angular/compiler-cli": "5.2.11",
  "@angular/core": "5.2.11",
  "@angular/forms": "5.2.11",
  "@angular/platform-browser": "5.2.11",
  "@angular/platform-browser-dynamic": "5.2.11",
  "@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"
},
```

Use of package.json

It's interesting…

- In **dependecies** section you can see the Project packages.
- In **devDependencies** you can find the packages that you have installed with the option **--save-dev** which are the packages used in development.

Test your App in your mobile phone…

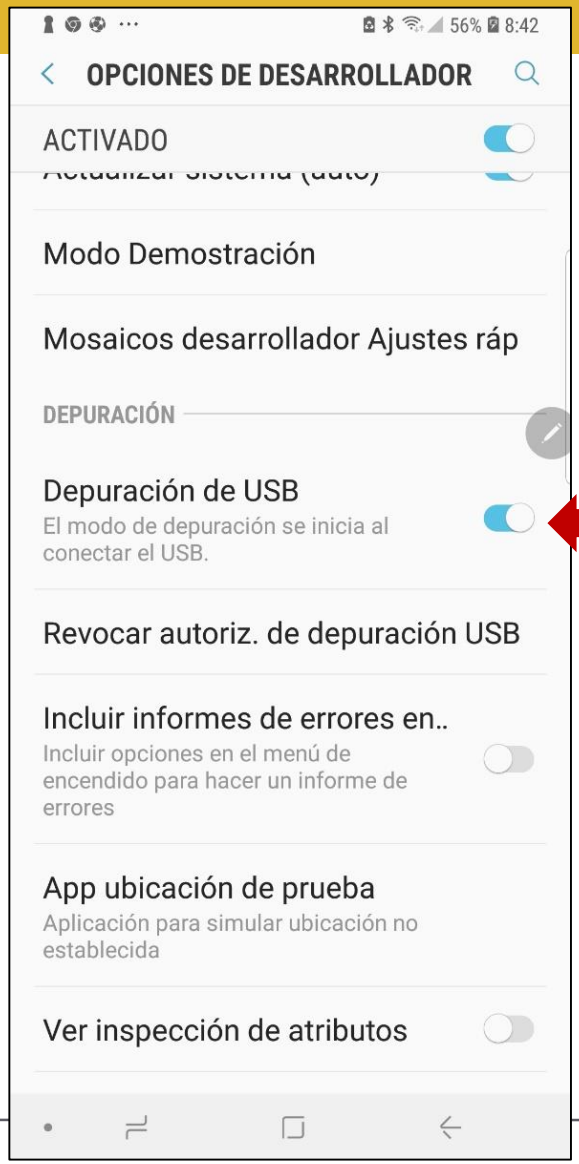


Test your App in your mobile phone …

In your mobile phone you must make visible the option "**Developer options**" which is hidden by default.

- In my mobile phone is in Settings → About the mobile phone → Software information → Compilation number
- About the option "**Compilation number**" You have to clic 7 times and then you'll see the option "Developper options"

Test your App in your mobile phone...



Test your App in your mobile phone ...

Once the option "**developper options**" is visible, then access to that option and enable the option "**USB debugging**"

Test your App in your mobile phone...

Test your App in your mobile phone…

(1) Connect your mobile phone via USB to your PC

```
$ ionic build
```

(2) This way you are creating your web project in the directory www. Note that the www folder appears in your project.

```
$ ionic cap add android
```

(3) The directory “Android” will be created, with your compiled Project inside.

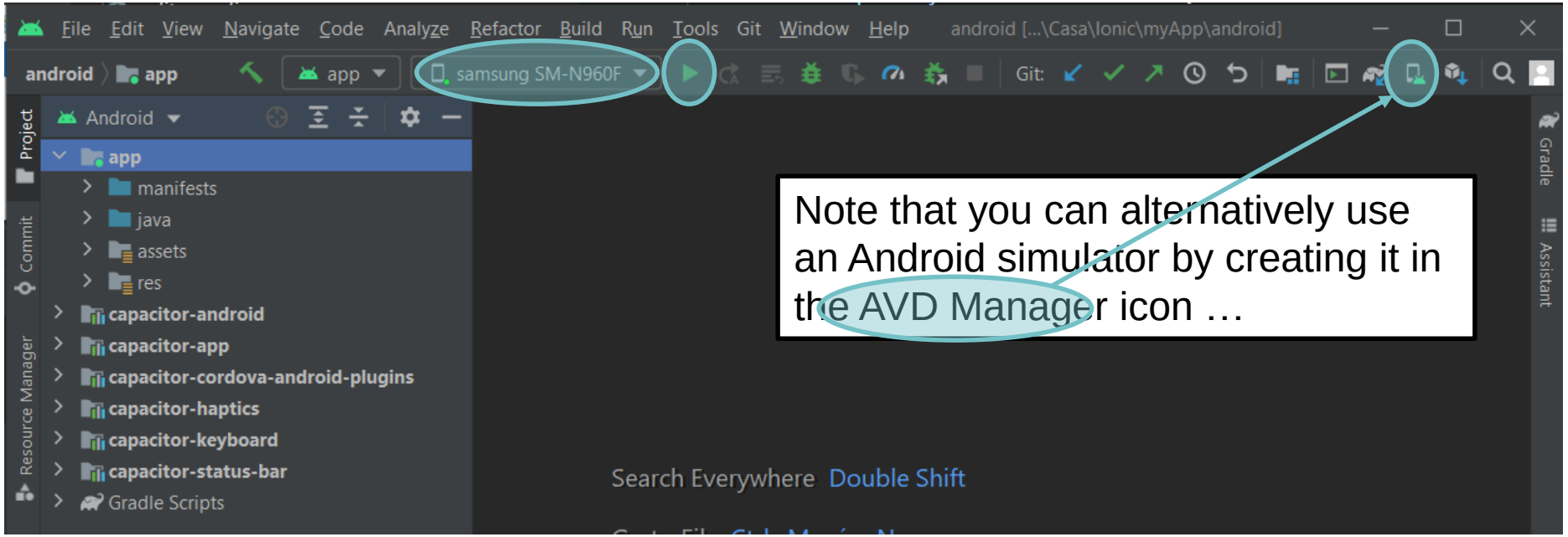
```
$ ionic cap open android
```

(4) This way your Android Studio will show your Android proyect

Test your App in your mobile phone...

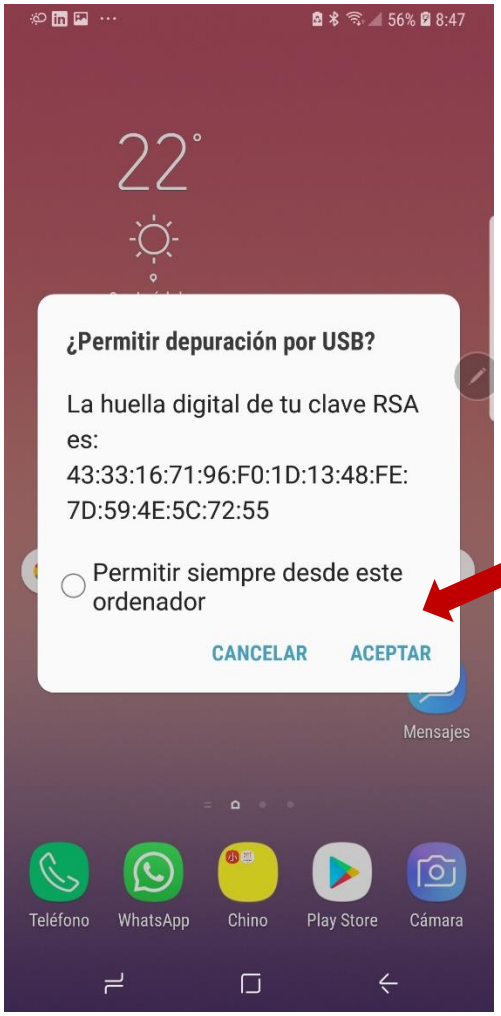
Test your App in your mobile phone ...

Now your project is natively open with the Android Studio IDE, and your mobile device name model should appear (in my case a Samsung SM-N960F) and you would only have to click on the green play icon to run your project on your mobile phone.



Note that you can alternatively use an Android simulator by creating it in the AVD Manager icon …

Test your App in your mobile phone...



Test your App in your mobile phone ...

When you clic on accept your app will finally show in your mobile phone.

… later if you make changes to the source code of your project…

Later if you change your code …

if you want to see them on your mobile then you must execute the following commands…

```
$ ionic build
```

(2) With this you are updating the content of your project in the www folder.

```
$ ionic cap copy
```

(3) The changes are copied to the different platforms. In this case to Android.

```
$ ionic cap sync
```

(4) With this command the projects of the different platforms will be synchronized. In our case so that you can see it from Android Studio.

Keep on learning...

Follow the next step-by-step example that is explained on the official Ionic website:

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

Be carefull with the version!

As of the creation of this tutorial, you should be working with version 5 of Ionic.

The Ionic 4 tutorials can still serve you mostly.

But the previous ones already have many differences.

Conclusions

What have we learned?

- We have simply installed the environment, creating an App and we have tested it on a mobile.
- We have also seen the most common possibilities to create mobile apps.

Next steps…

- We have used Angular as a framework. So the next thing is to start working with Angular on Ionic.