
React Native

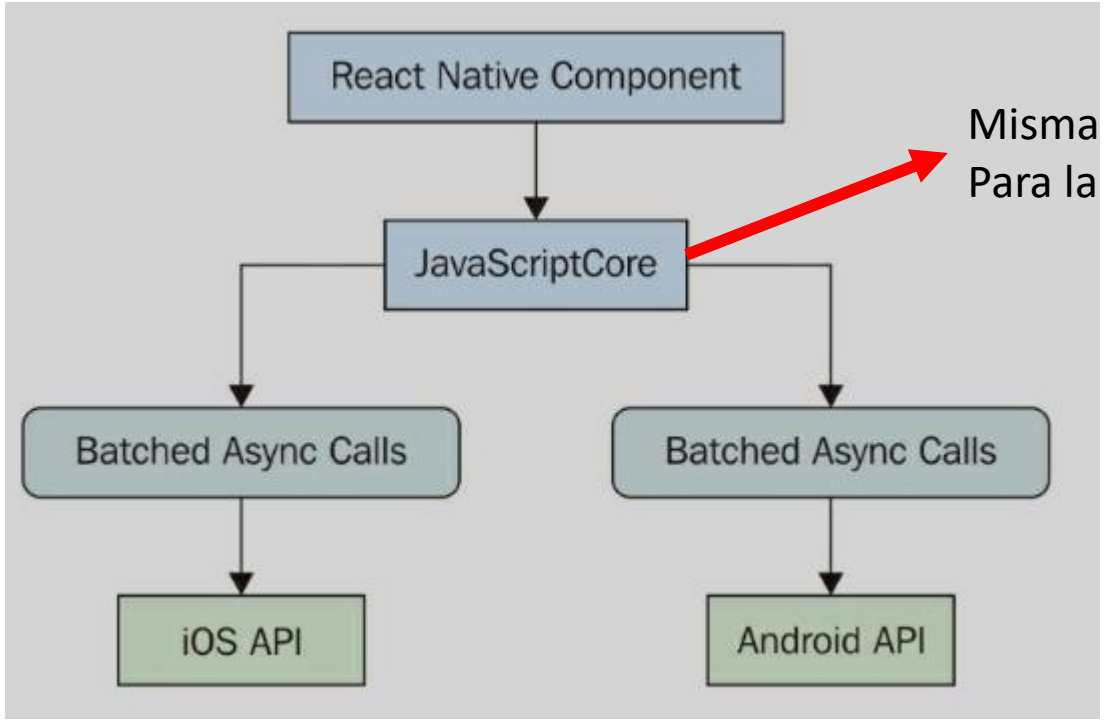
1era CLASE

INSTALACION ENTORNO DESARROLLO

INTRODUCCION

QUE ES??

- Es un esfuerzo de Facebook para reusar React en la creación de apps nativas para móviles.
- React y JSX son muy buenos declarando UI componentes y tienen gran demanda para las apps, tiene sentido usar lo aprendido para las webApps
- Trata de implementar componentes comunes entre IOS y Android

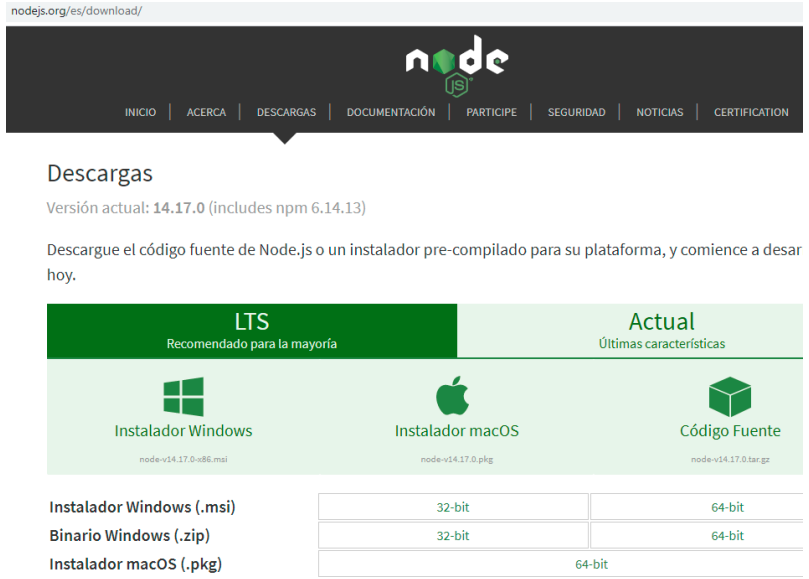


Uber Eats

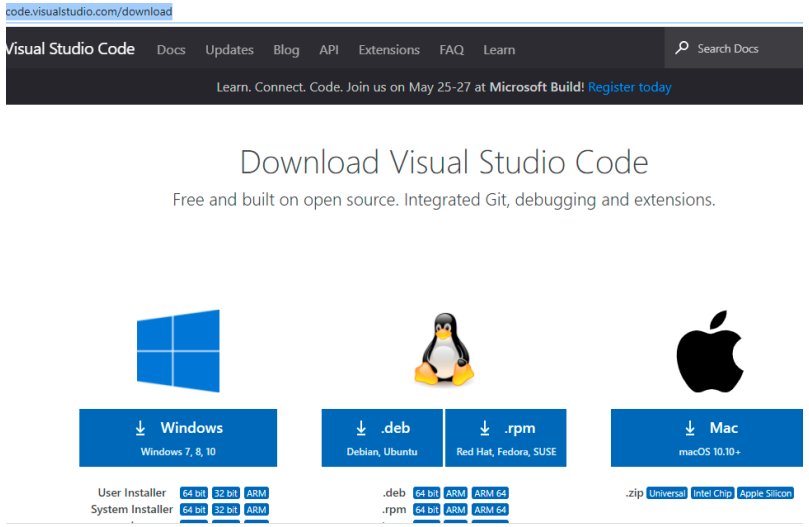
Sin REACT NATIVE

SOFTWARE a usar

- NODE JS
  - https://nodejs.org/es/download/



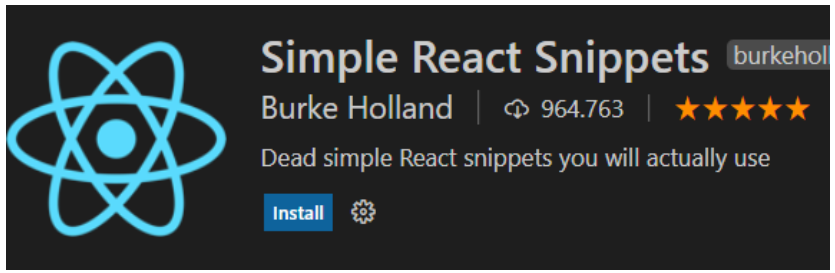
- VISUAL SOURCE CODE
  - https://code.visualstudio.com/download
  - EXTENSIONES



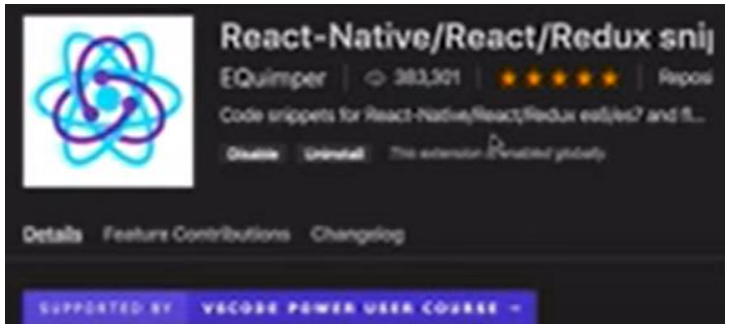
VISUAL STUDIO CODE

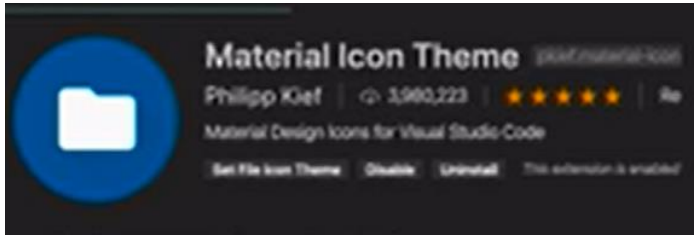
Top 10 Extensions for ReactJs in VSCode are:

- The ***Bracket Pair Colorizer*** extension.
- The ***change-case*** extension.
- The ***Code Spell Checker*** extension.
- The ***Duplicate Selection*** extension.
- The ***EditorConfig for VS Code*** extension.
- The ***VSCode React Refactor*** extension.
- The ***npm Intellisense*** extension.
- The ***ESLint*** extension.
- The ***ES7 React/Redux/GraphQL/React-Native snippets*** extension.
- The ***Prettier – Code formatted*** extension.



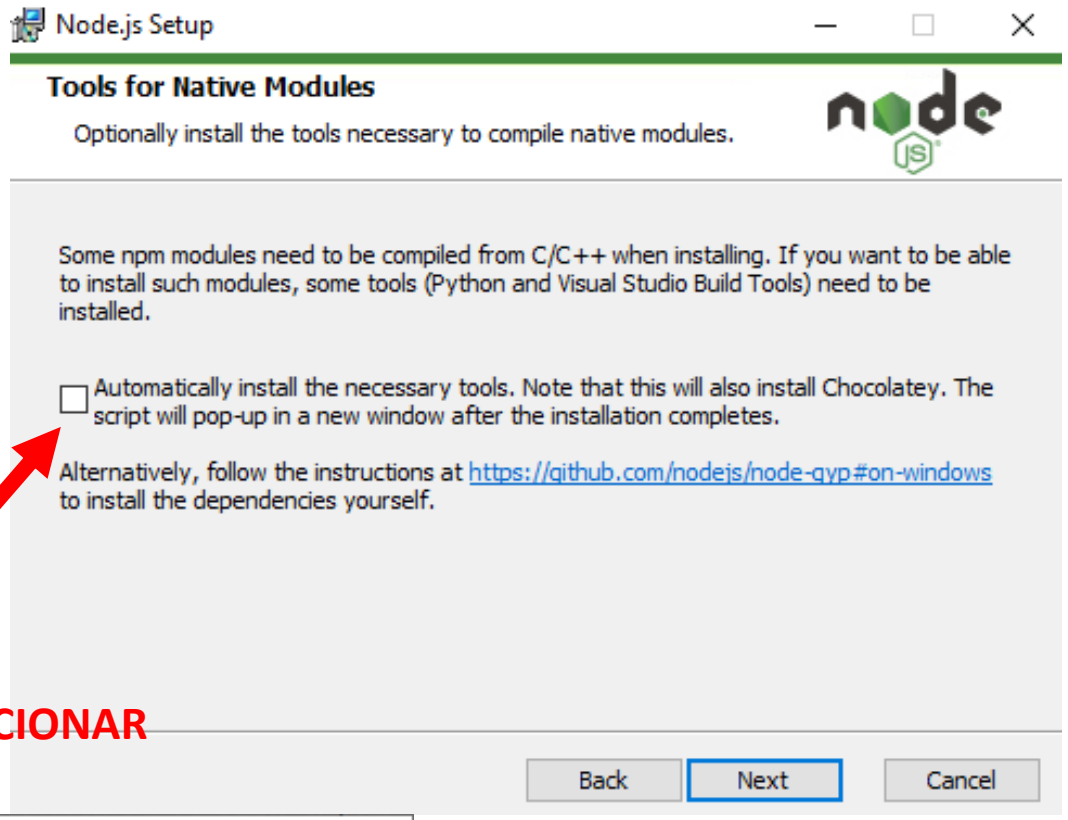
Bracket pair colorizer
Auto rename tag



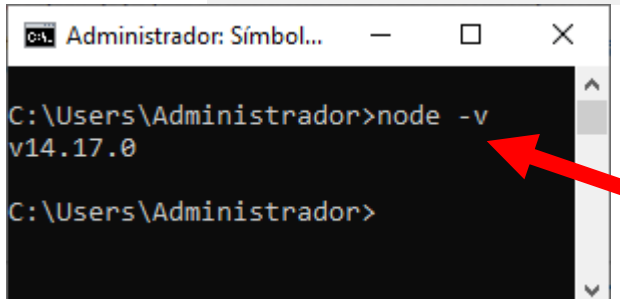


NPM= node package manager

NODE JS



NO SELECCIONAR



Verificar la instalación de NODE

Instalar el cliente REACT

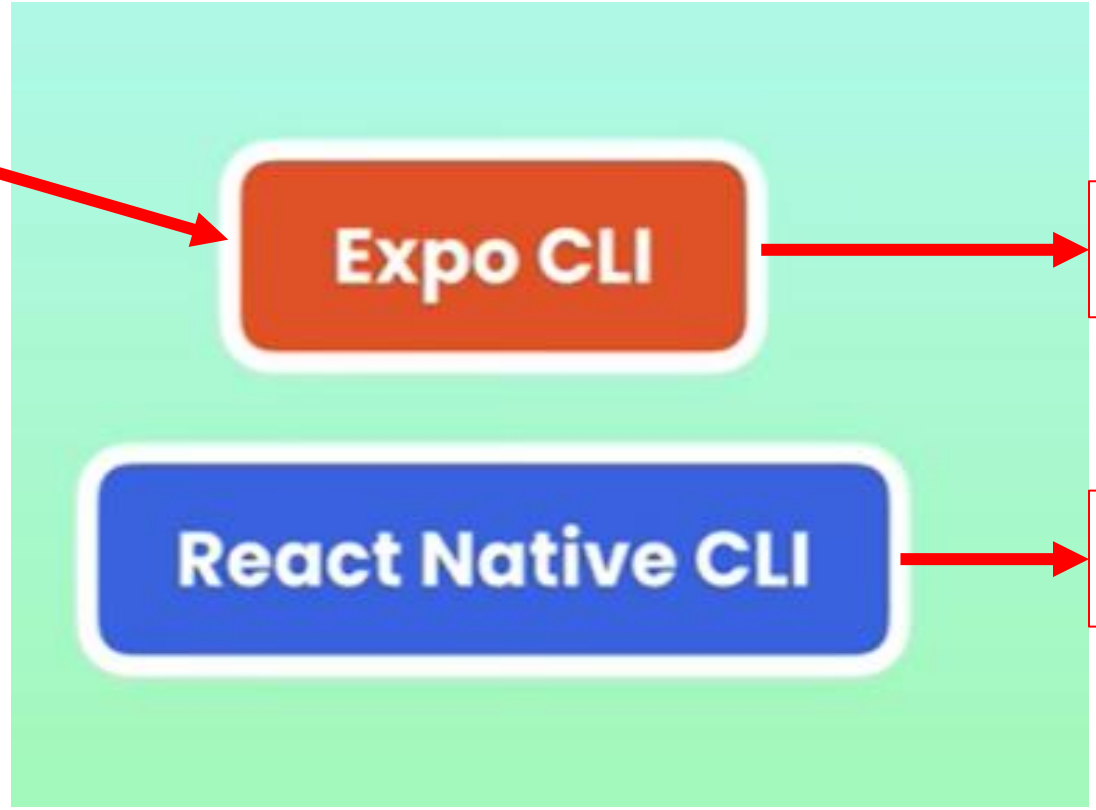
- Abrir una ventana CMD
- Tipear
  - npm install –g create-react-app
  - Y luego
  - create-react-app **teoria**
  - 0 npx create-react-app **teoria**

- Ir dentro de ese directorio (**cd teoria**) y
  - **npm start**

BOOTSTRAP
Dentro del directorio del proyecto
npm i bootstrap@4.1.1

Para crear React Apps tenemos estas 2 opciones

Usamos esta

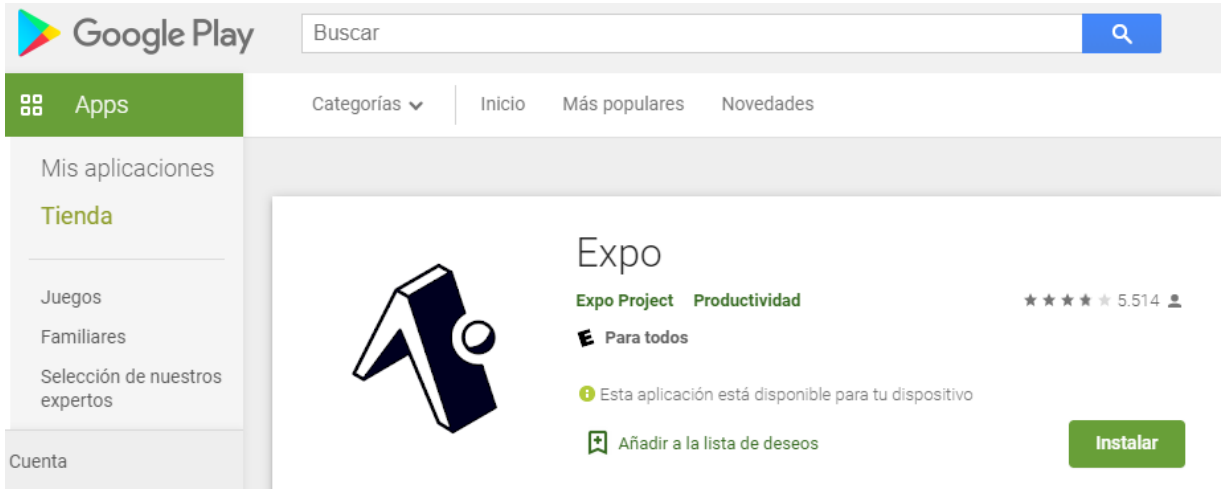


INSTALAR
npm install –g expo-cli

INSTALAR
npm install react-native-cli -g

HOLA MUNDO

- Chequear la versión de node , superior a 12
  node –v
- **npm i –g expo-cli**
- **expo Client en el celu …**



EXTENSIONS en VS

- react native tools
- react-native/react/redux
- prettier
- Material icon Theme

- expo init HolaMundo

Habilitar en el celu la opción de DESARROLLADOR

Entra en los ajustes del móvil

Toca en Información del dispositivo

Toca siete veces en Número de compilación

Si fuera necesario, introduce el PIN

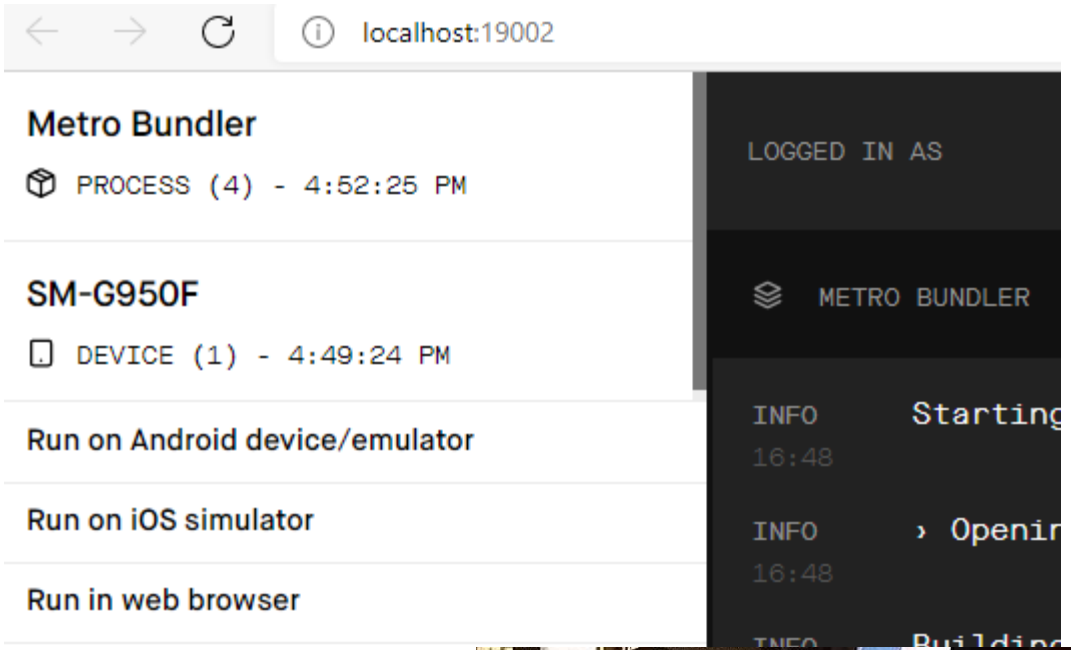
HOLA MUNDO

en una ventada CMD, ejecutar

1. Chequear la versión de NODE.JS …. Superior a 12
2. **npm install –g expo-cli**
3. expo init holaMundo 0
   npx créate-expo-app -- template
4. elegir Managed workflow …… es el default
5. tipear el nombre para la app….. holaMundo

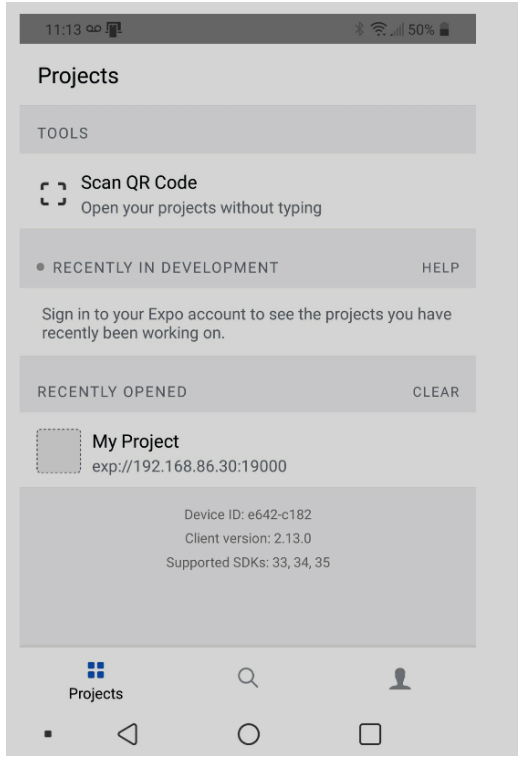
Comprobar la app en el celular

1. En una ventana CMD, ir al directorio del proyecto
   cd …./ holaMundo
   npm start
2. ir a un navegador …. localhost:19002
3. **Conectar el celular a la compu**
4. Elegir en Connection … LOCAL
5. Click en Run on Android device

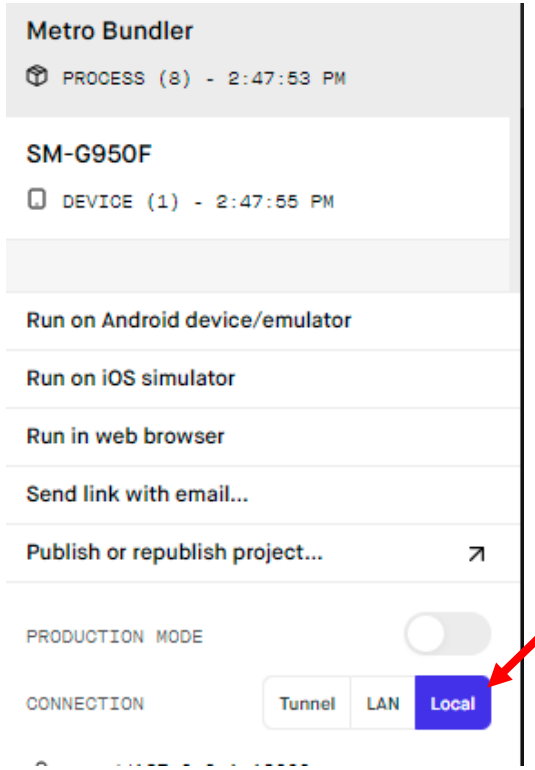


HOLA MUNDO

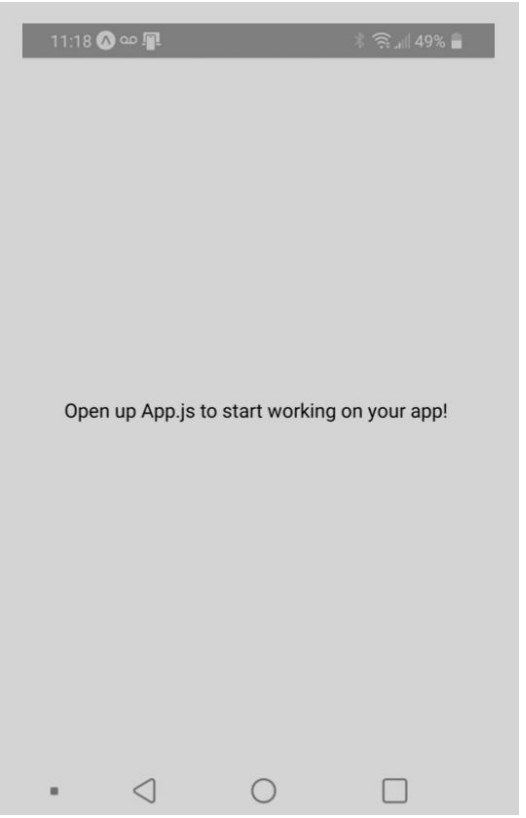
- Para ver la app en tu celular, bajar de PlayStore la **Expo** app
- Una vez instalada, escanear el QR



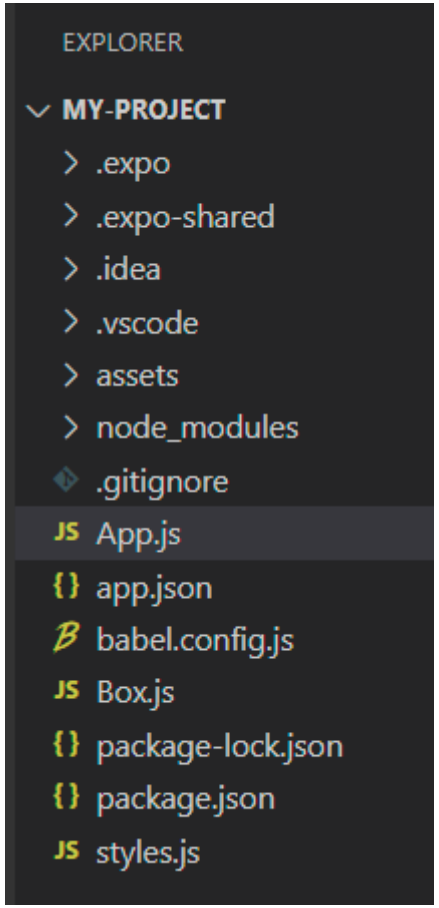
En el browser..
Aparece tu celu



tu celu

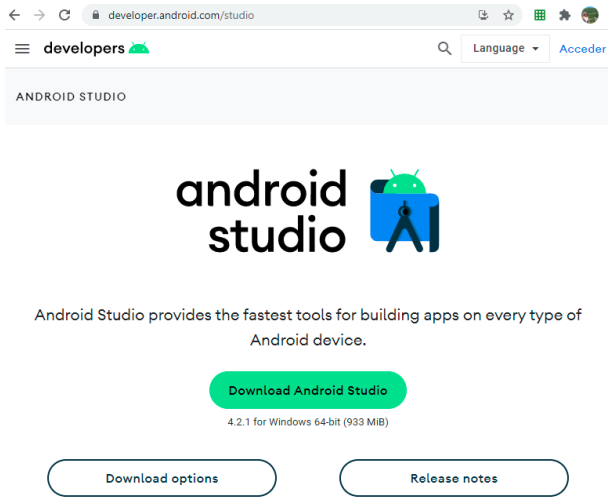


Ejemplo HOLA MUNDO



ANDROID STUDIO

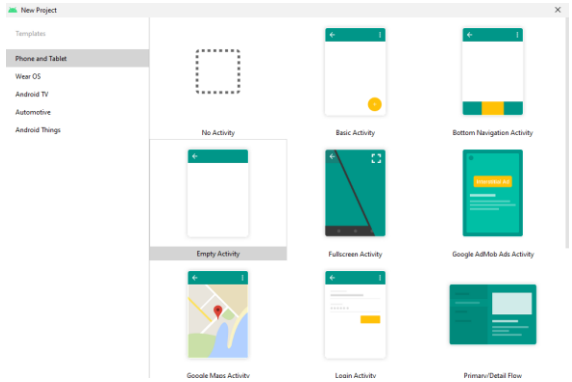
- Para ver la app en un simulador
- IOS …. **Xcode**
- ANDROID
- OPCION1 … download Android studio
  - https://developer.android.com/studio
  - O ….
  - https://docs.expo.io/workflow/android-studio-emulator/
- OPCION2 …
- Snack - React Native in the browser (expo.io)
- https://snack.expo.io/

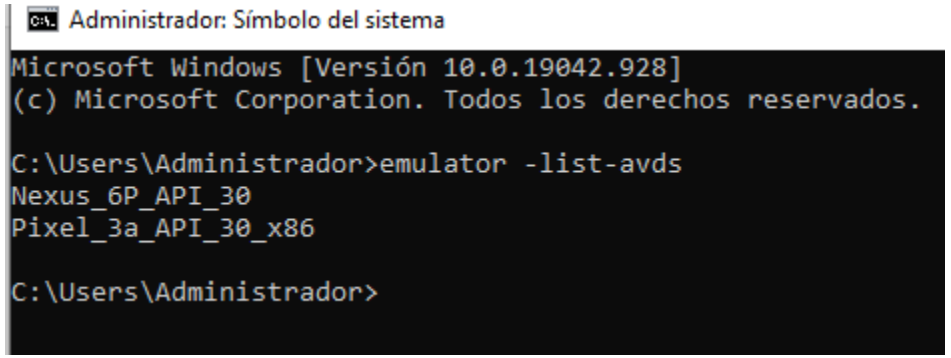


ANDROID STUDIO

Pueden tener problemas con la virtualización de Windows …

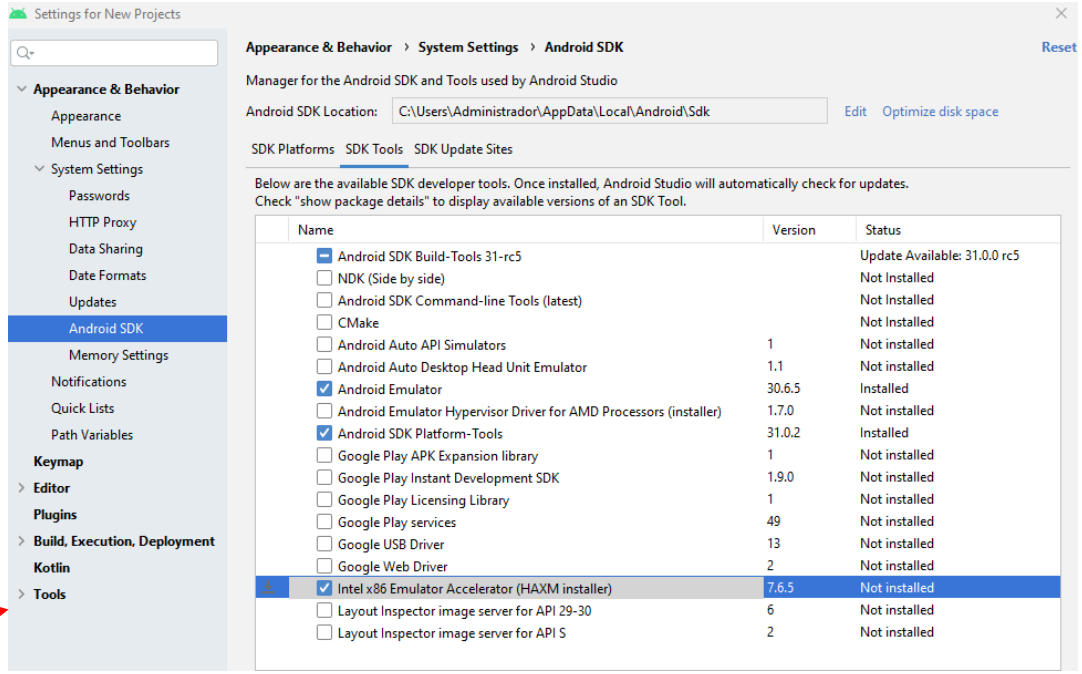
Error de HAXM





Y no poder ejecutar el emulador

Hay que modificar la BIOS de la compu



https://www.minitool.com/news/enable-virtualization-windows-10.html

ANDROID STUDIO

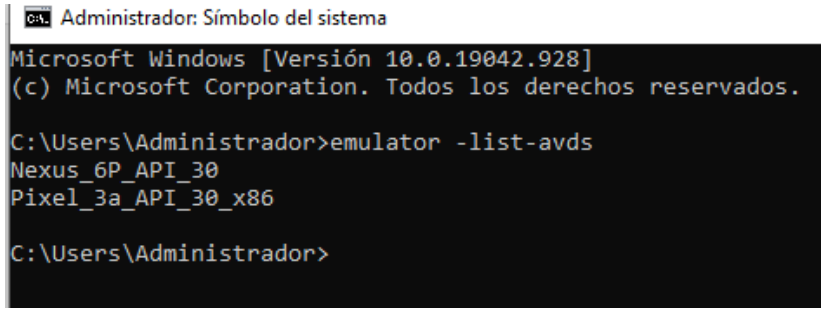
¿Cómo iniciar su emulador con comandos cli?
(omita si planea iniciar usando AVD Manager)

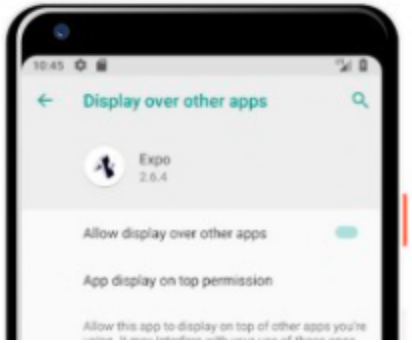
1. Ejecute el comando " **emulator -list-avds"** . Esto obtendrá una lista de emuladores disponibles que creó.
2. Luego ejecute el comando " **emulator -avd [su nombre avd]"** para iniciar su emulador. Reemplace **[su nombre avd]** por el nombre que aparece en el " **emulator -list-avds"** del paso 1.

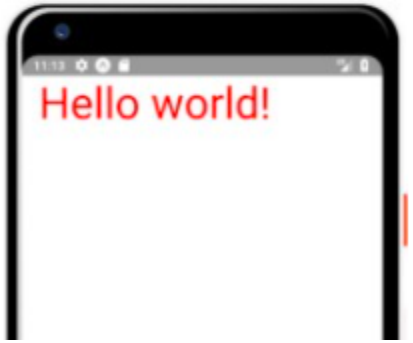
```
kongwenyao@wnyao-macbook: ~ $ emulator -list-avds
Android_Accelerated_Nougat
Pixel_2_API_27
Pixel_2_XL_API_P
Pixel_C_API_P
kongwenyao@wnyao-macbook: ~ $ emulator -avd Pixel_2_XL_API_P
```

Lanzamiento del emulador de Android.

1. Asegúrese de estar en la raíz de su proyecto y de que su emulador se esté ejecutando
2. Ejecute el comando " **react-native run-android"**
3. Si es la primera vez que ejecuta un flujo de trabajo administrado, el emulador mostrará el mensaje **"Permitir dibujar sobre otras aplicaciones"** . Simplemente continúe presionando **ok** .
4. Proceda a permitir que la aplicación se muestre sobre otras aplicaciones **alternando el botón** . Esto funciona tanto con la aplicación cliente de Expo (lo llamamos flujo de trabajo de gestión) o sin ella.
5. ¡Hurra! Ahora, la **aplicación React Native se** ejecuta en un emulador de Android **.**







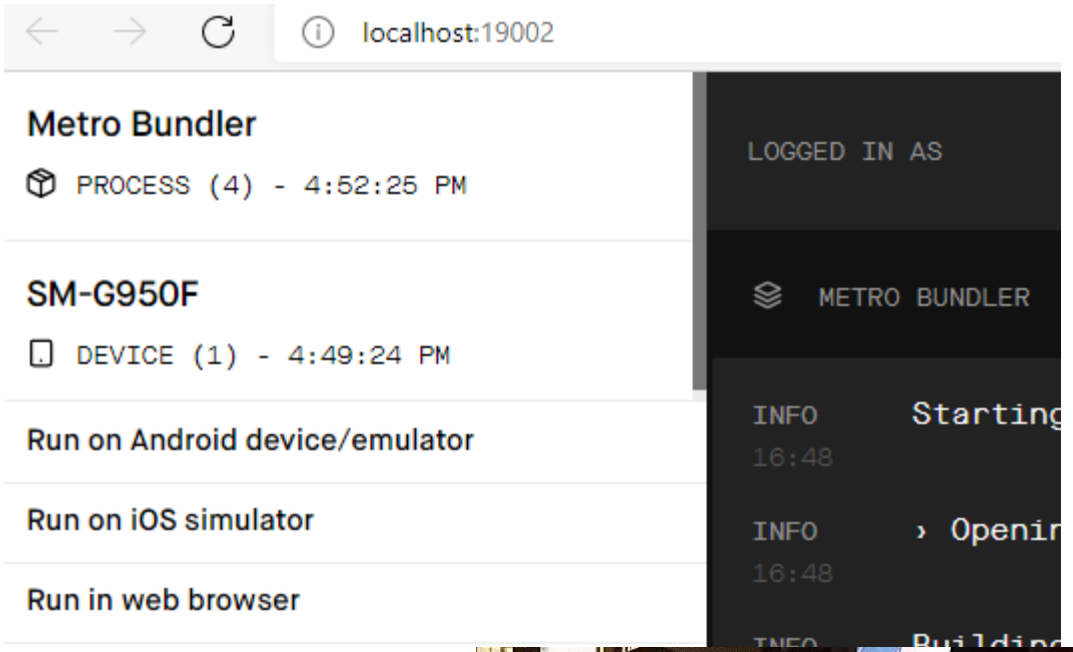
Conceptos Fundamentales

en una ventada CMD, ejecutar

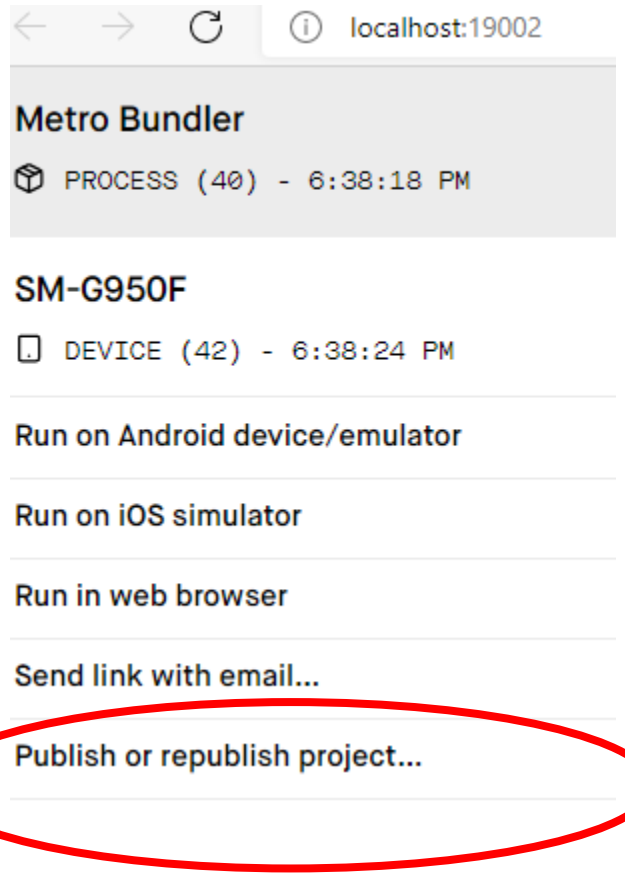
1. Chequear la versión de NODE.JS …. Superior a 12
2. **npm install –g expo-cli**
3. expo init **DoneWithIt**
4. elegir Managed workflow …… es el default
5. tipear el nombre para la app….. **DoneWithIt**

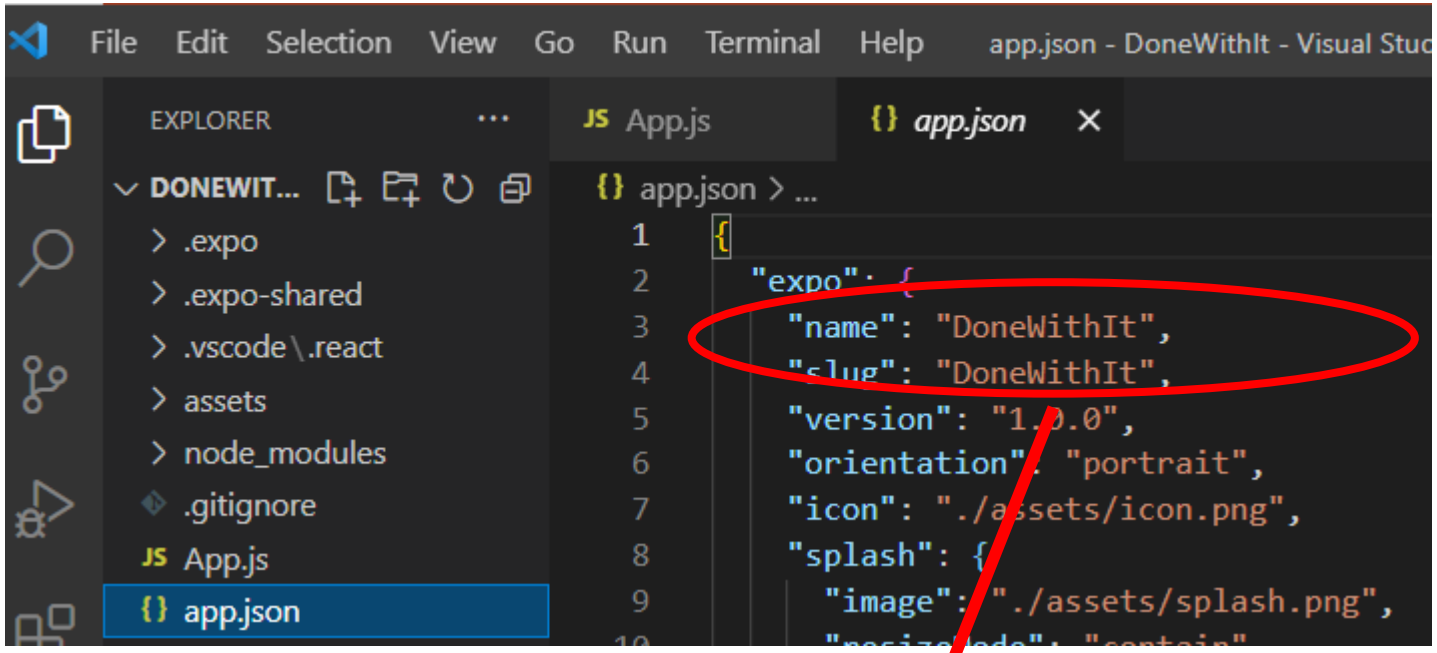
Comprobar la app en el celular

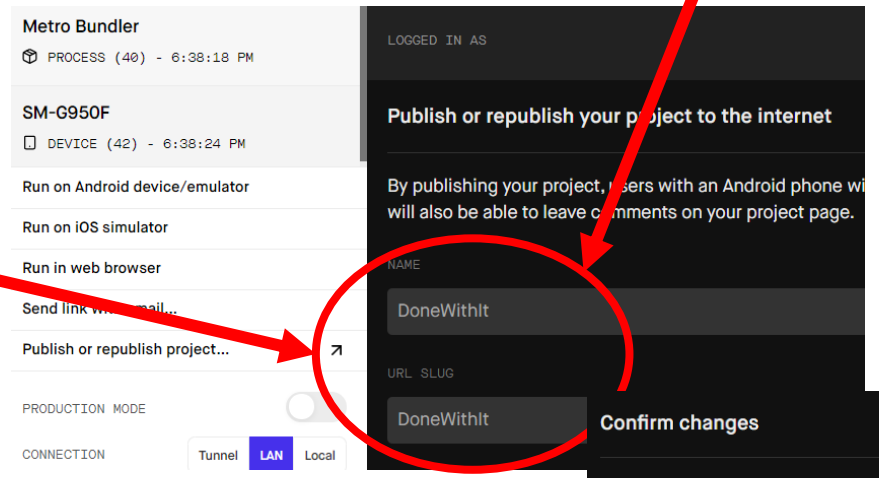
1. En una ventana CMD, ir al directorio del proyecto cd …./**DoneWithIt**
2. **npm start**
3. ir a un navegador …. localhost:19002
4. Conectar el celular a la compu
5. Elegir en Connection … LOCAL
6. Click en Run on Android device

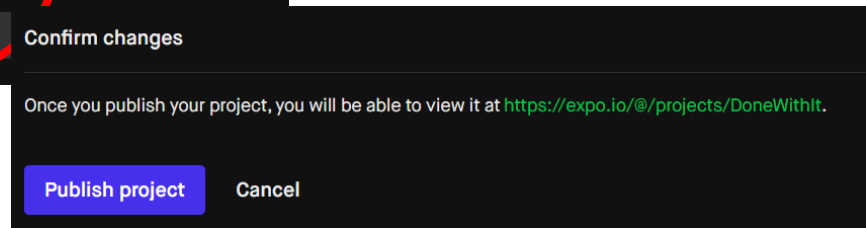


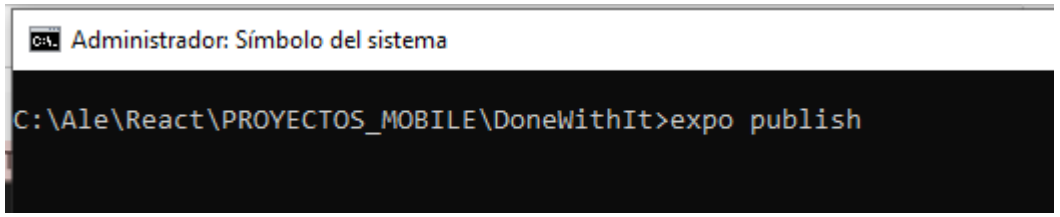
Publishing


localhost:19002
Metro Bundler
PROCESS (40) - 6:38:18 PM
SM-G950F
DEVICE (42) - 6:38:24 PM
Run on Android device/emulator
Run on iOS simulator
Run in web browser
Send link with email...
Publish or republish project...

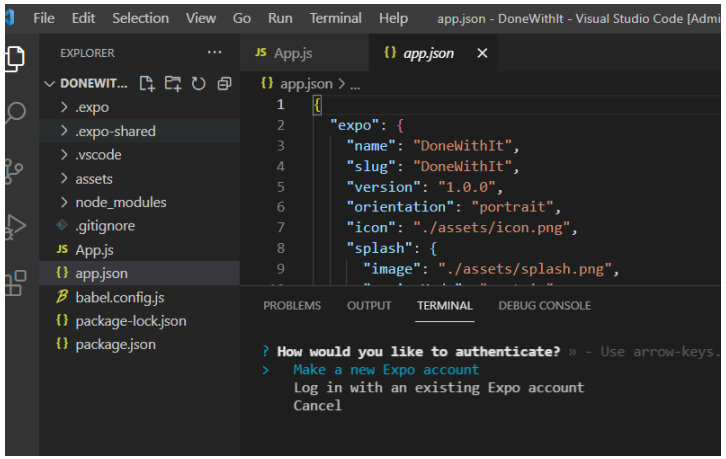

File Edit Selection View Go Run Terminal Help
app.json - DoneWithIt - Visual Stud
EXPLORER
App.js
app.json
DONEWIT...
.expo
.expo-shared
.vscode\.react
assets
node_modules
.gitignore
App.js
app.json
app.json > ...
{
"expo": {
"name": "DoneWithIt",
"slug": "DoneWithIt",
"version": "1.0.0",
"orientation": "portrait",
"icon": "./assets/icon.png",
"splash": {
"image": "./assets/splash.png",

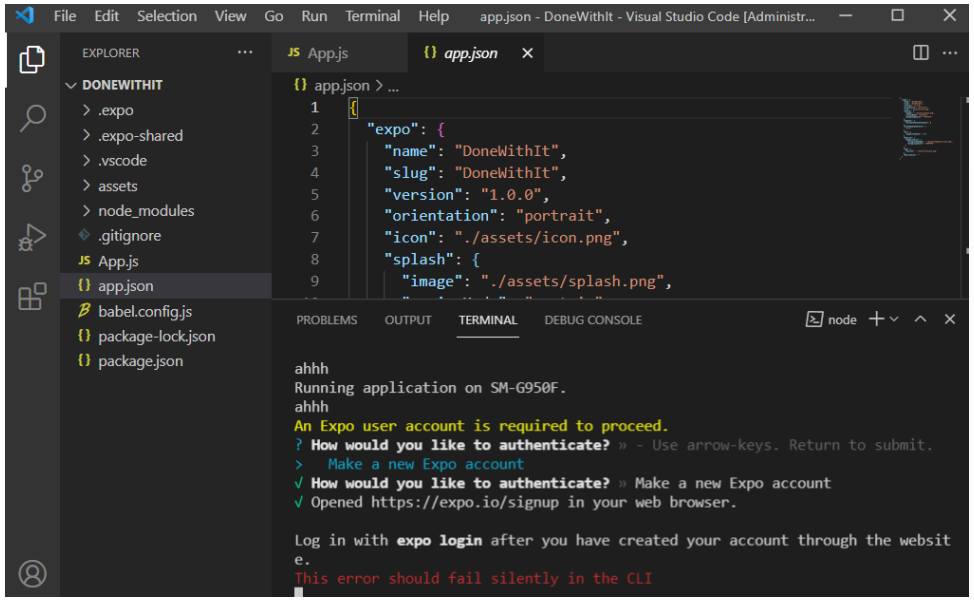

Metro Bundler
PROCESS (40) - 6:38:18 PM
SM-G950F
DEVICE (42) - 6:38:24 PM
Run on Android device/emulator
Run on iOS simulator
Run in web browser
Publish or republish project...
PRODUCTION MODE
CONNECTION
Tunnel
LAN
Local
LOGGED IN AS
Publish or republish your project to the internet
By publishing your project, users with an Android phone will also be able to leave comments on your project page.
NAME
DoneWithIt
URL SLUG
DoneWithIt

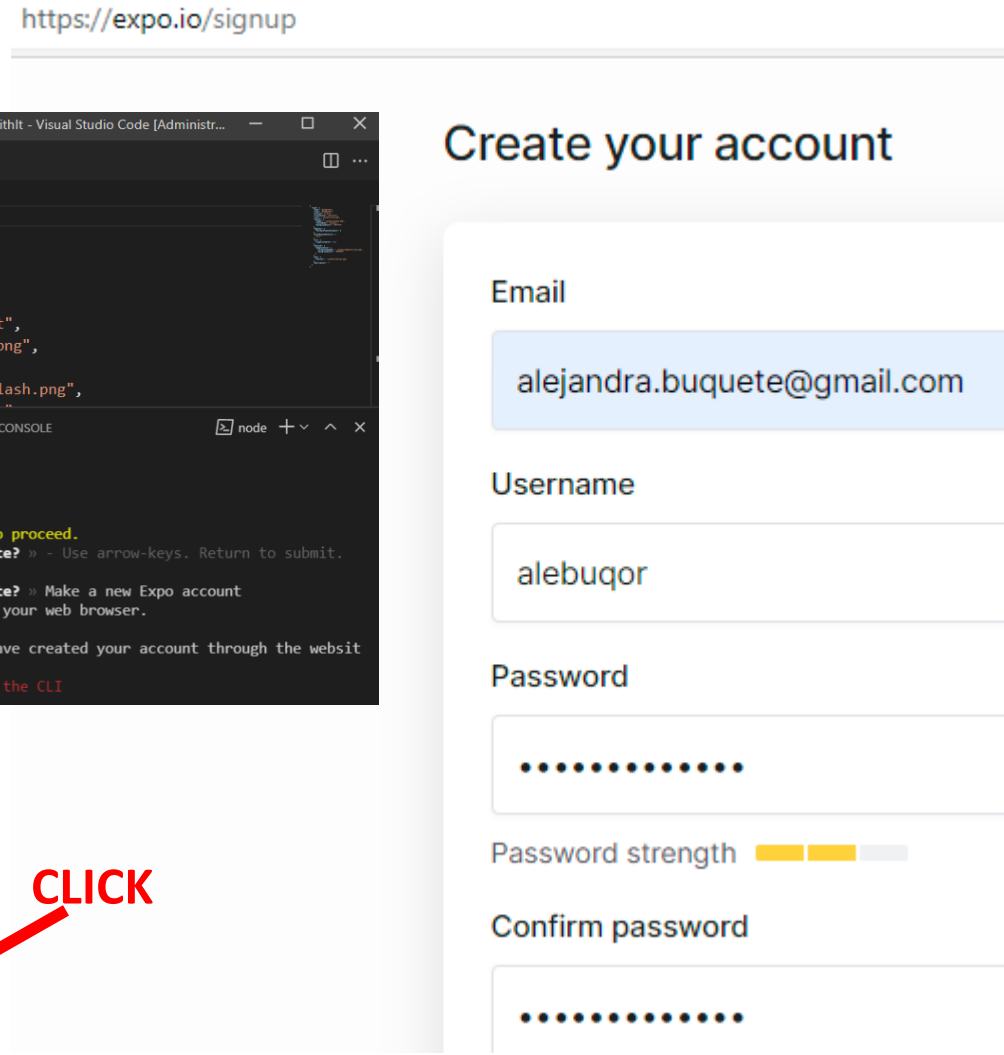

Confirm changes
Once you publish your project, you will be able to view it at https://expo.io/@/projects/DoneWithIt.
Publish project
Cancel

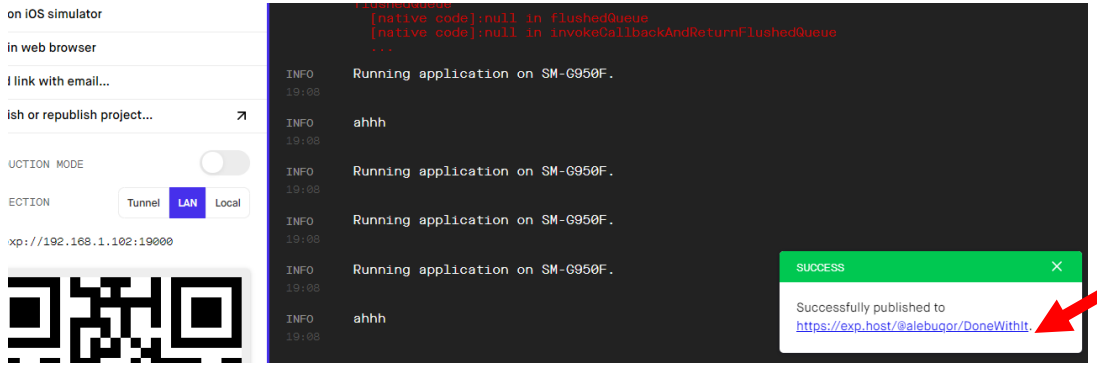

Administrador: Símbolo del sistema
C:\Ale\React\PROYECTOS_MOBILE\DoneWithIt>expo publish

Crear un usuario en expo …

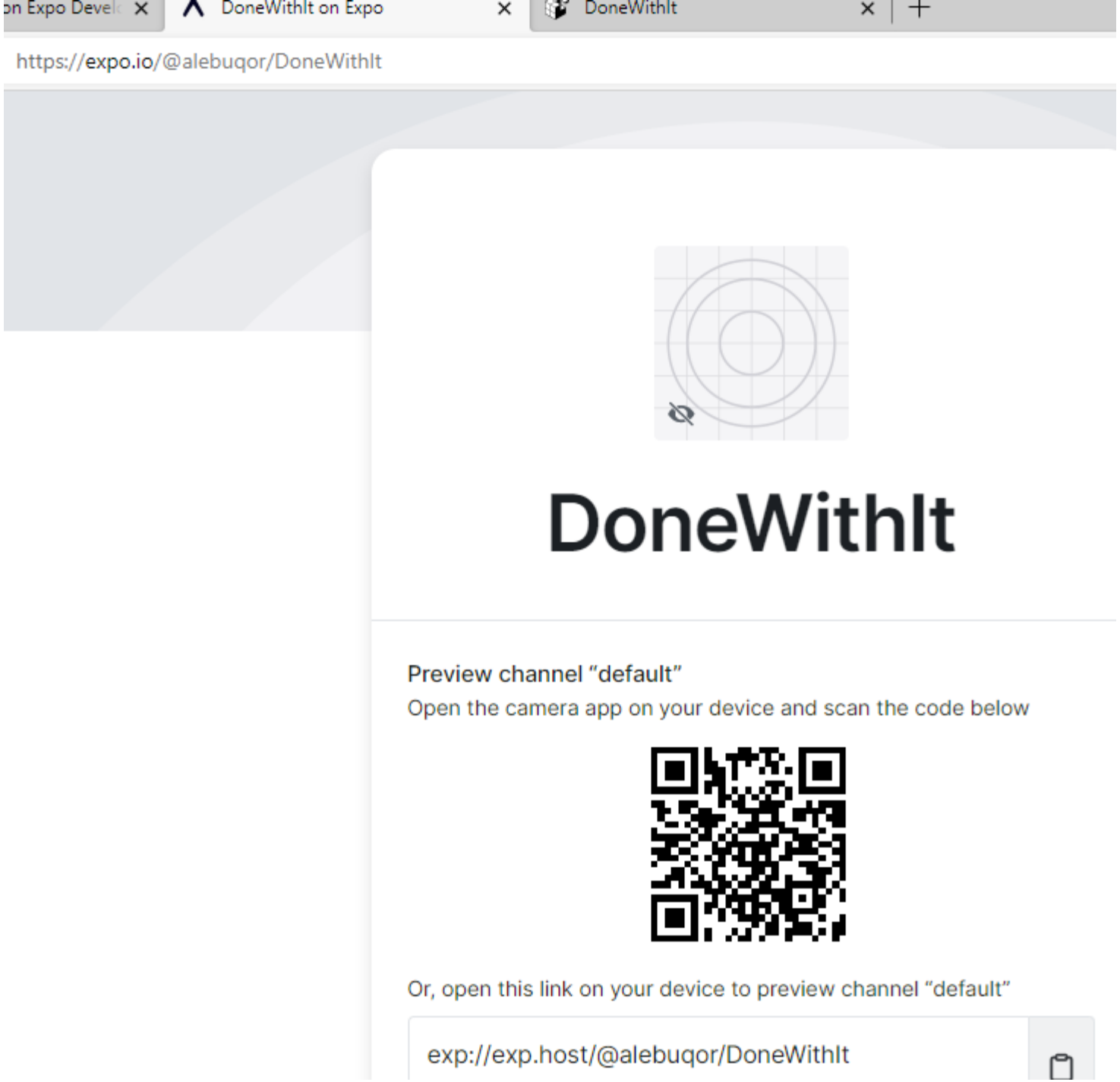








CLICK

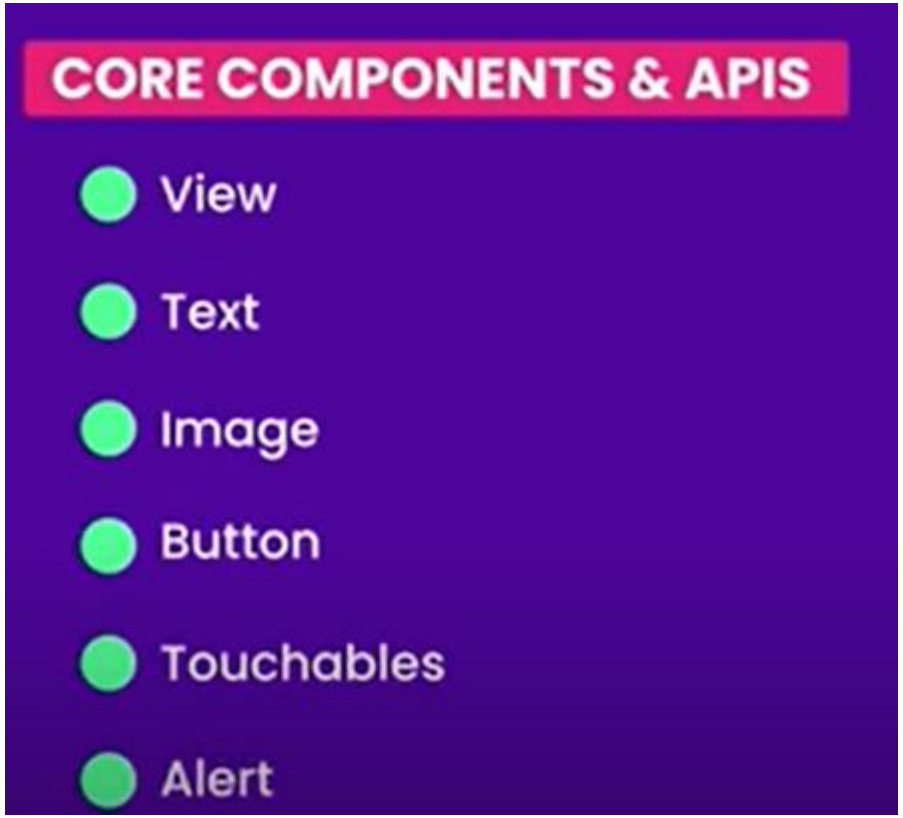

on Expo Devel
DoneWithIt on Expo
DoneWithIt
https://expo.io/@alebuqor/DoneWithIt
DoneWithIt
Preview channel "default"
Open the camera app on your device and scan the code below
Or, open this link on your device to preview channel "default"
exp://exp.host/@alebuqor/DoneWithIt

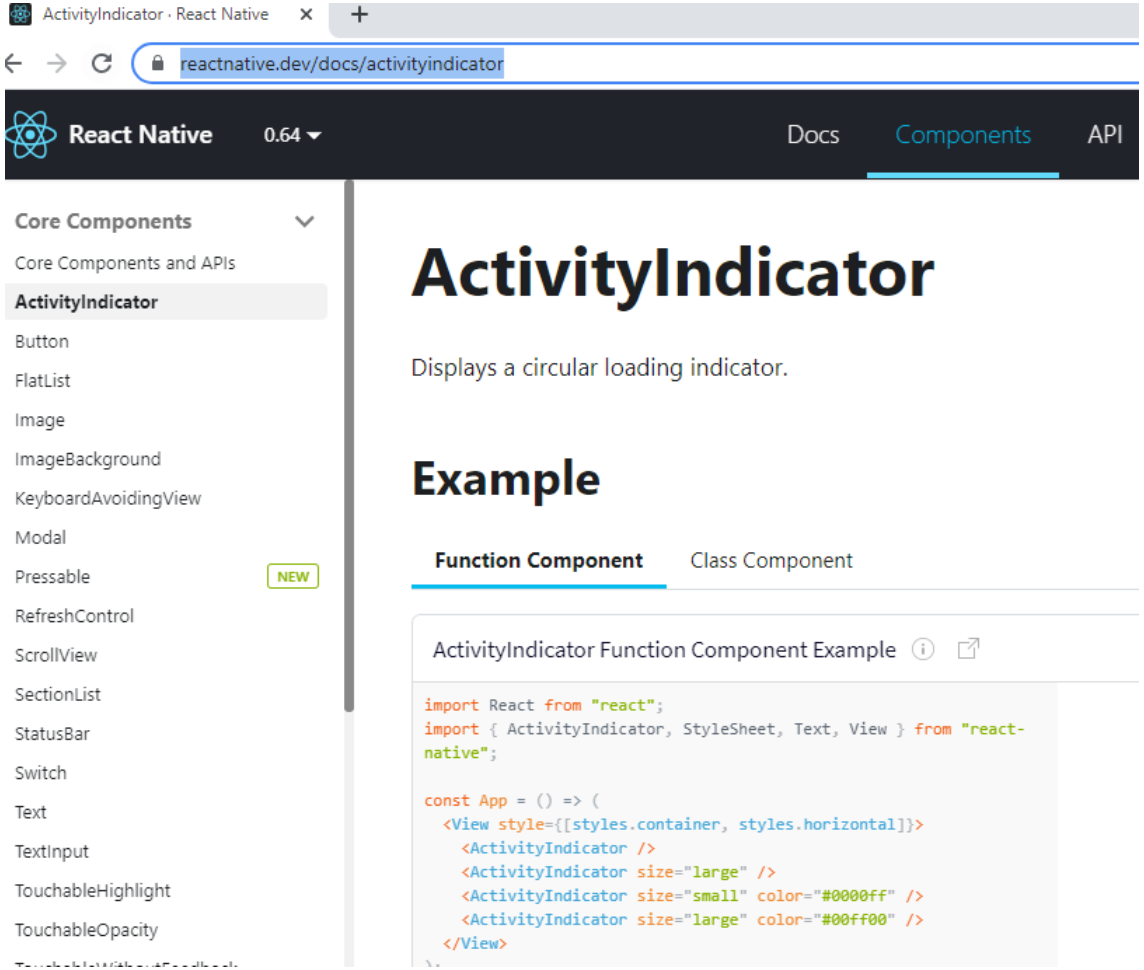
2da CLASE

ORQUESTACION GRAFICA

INTRODUCCION

https://reactnative.dev/docs/activityindicator



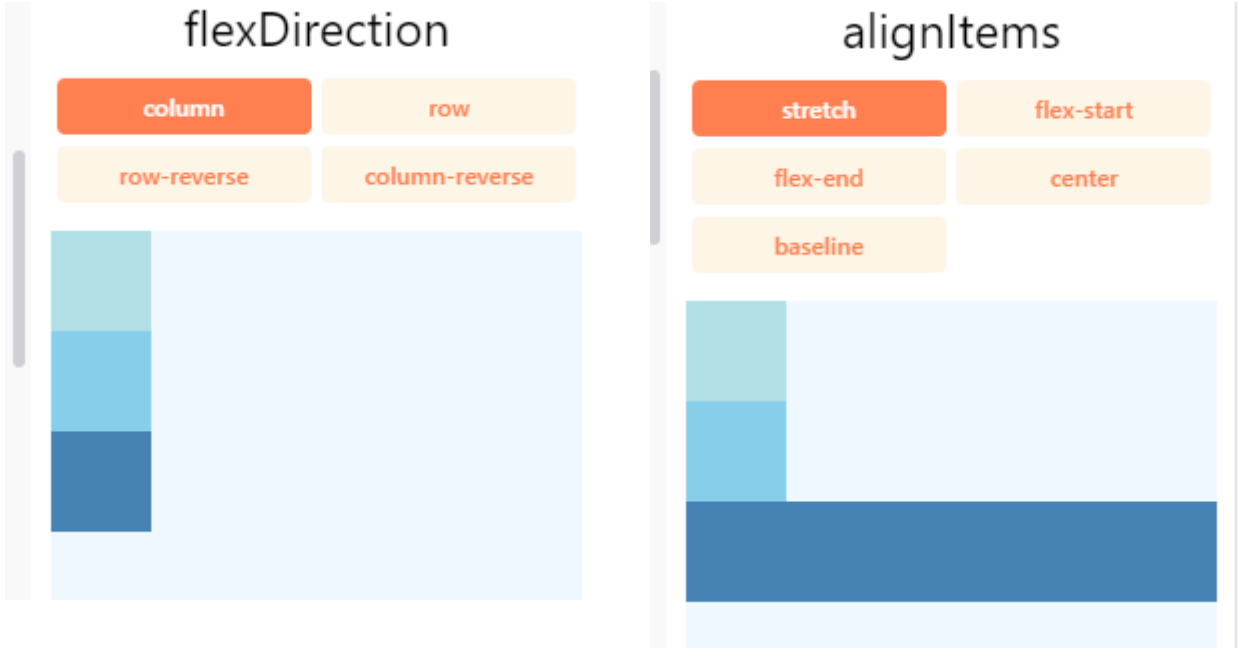


Lo usamos en App.js

- Es un container component,
- Es como un DIV
- Se usa para agrupar o colocar componentes hijos
- <View style={styles.container}>
  - styles
    - Podemos usar para manejar colores
      - Rgb, nombres o hex
      - backgroundColor: '#fff‘ ….BLANCO

- Usamos el tag
- <Text> …. No escribimos texto asi nomas
- https://reactnative.dev/docs/text

Flexbox



Usa una combinación de

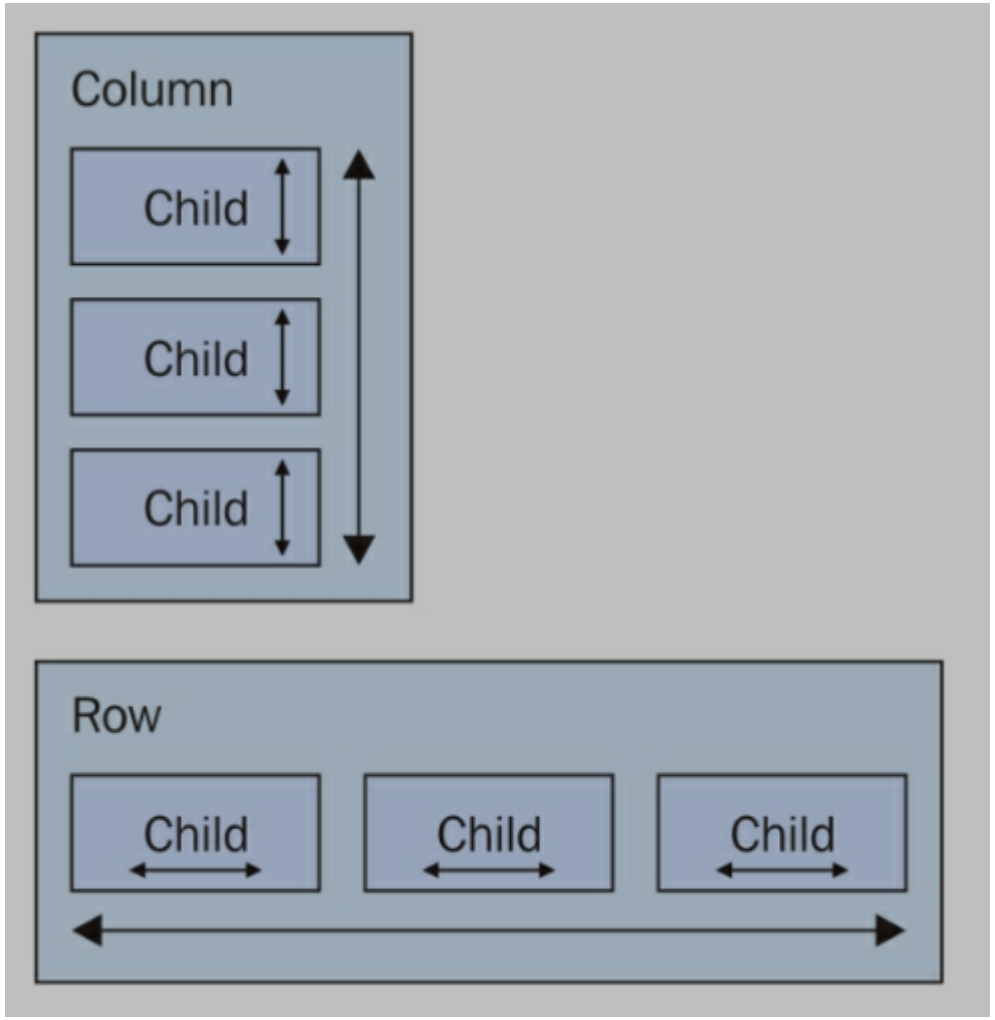
- flexDirection
- alignItems
- justifyContent

Define COMO los ítems llenaran el espacio disponible

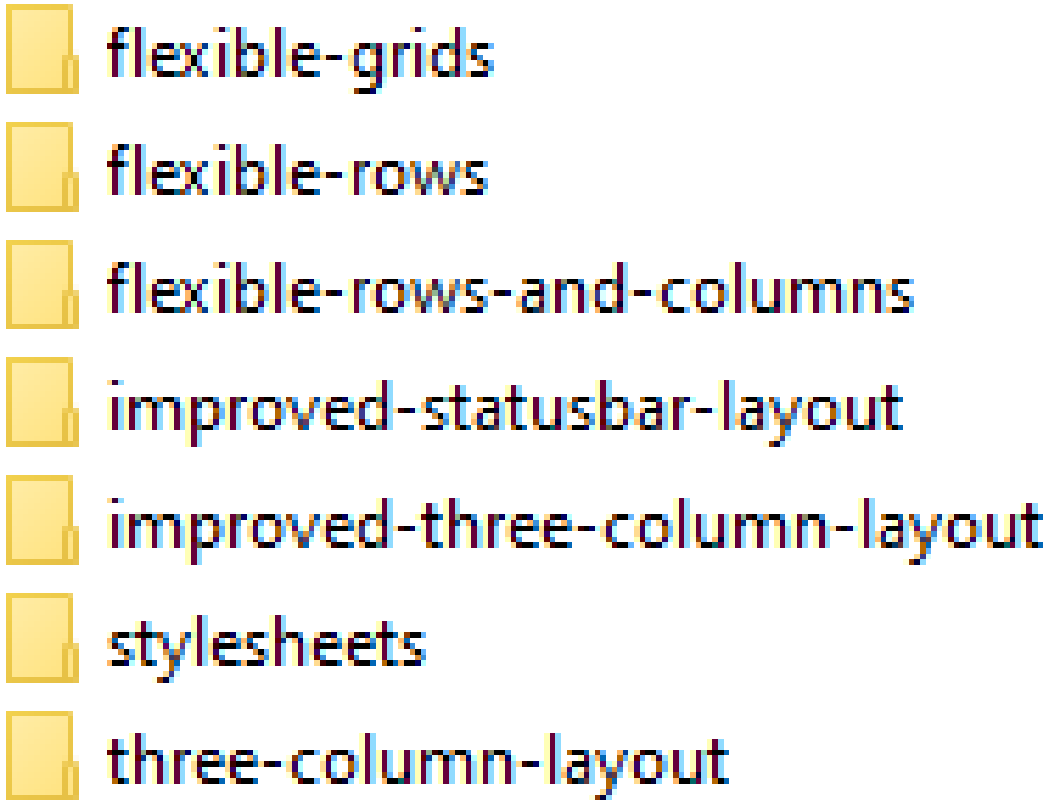
Building Responsive Layouts with Flexbox

Flexbox

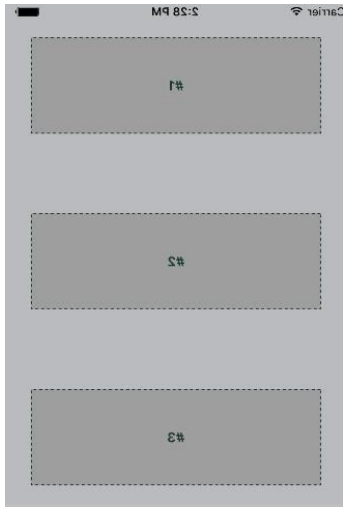
- Es una caja simple que actua como contenedor y posee elementos que se comportan como hijos
- Ambos son flexibles cuando se renderizan en la pantalla
- Tienen una dirección
  - Arriba/abajo (columna)
  - Izq/der (fila)



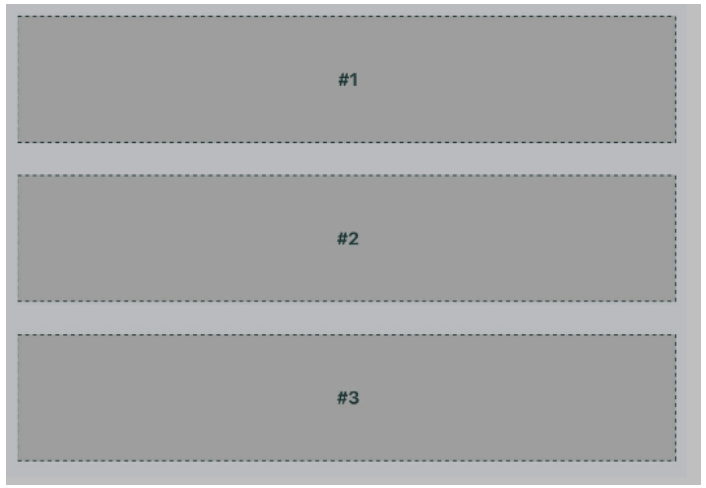
Building Responsive Layouts with Flexbox



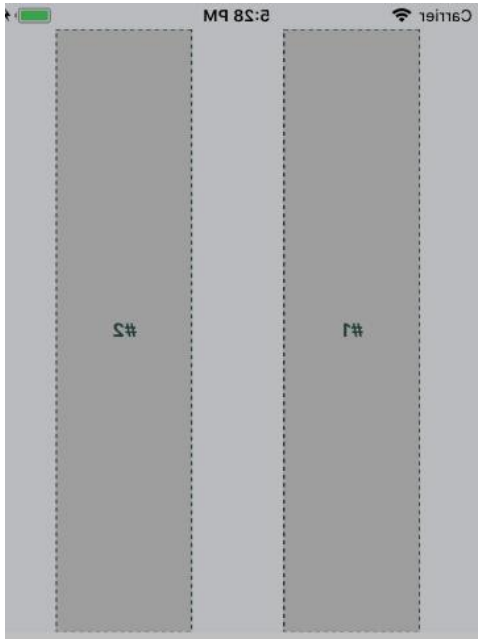
three-column-layout



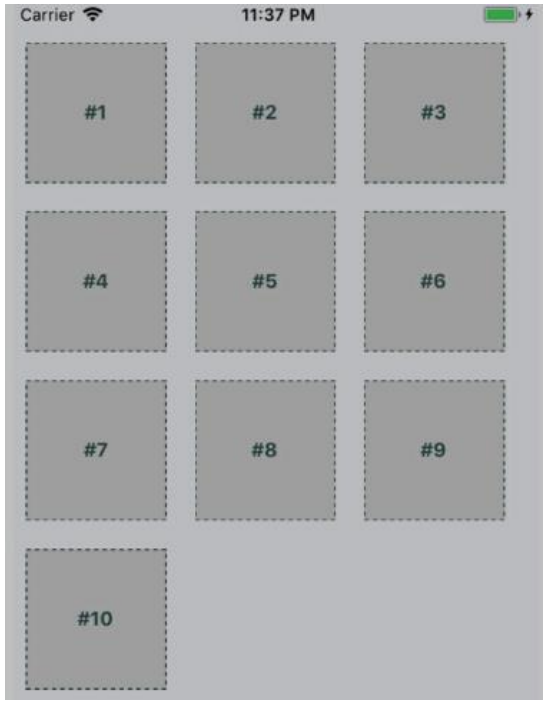
improved-three-column-layout



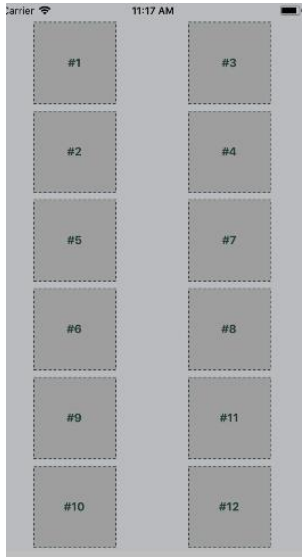
flexible-rows



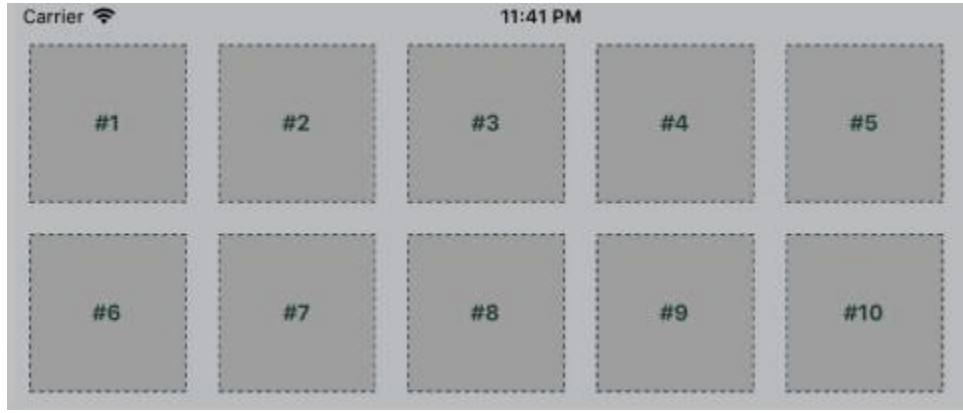
flexible-grids



flexible-rows-and-columns



Landscape orientation con la flexible-grid



Image

- import { StyleSheet, Text, View, **Image** } from 'react-native';

LOCALES

```
<Image source={require('./assets/favicon.png')}/>
```

DE INTERNET

```
<Image source={{
    width:200,
    height:300,
    uri: 'https://picsum.photos/id/1/200/300'}}/>
```

Touchables

<**TouchableWithoutFeedback** onPress={()=>console.log("imagen presionada")}>

<**TouchableOpacity** onPress={()=>console.log("imagen presionada")}>

<**TouchableHighlight** onPress={()=>console.log("imagen presionada")}>

TouchableNativeFeedback ➔ solo para Android

No funciona exactamente para imágenes, si para <views>

```
/*
<TouchableNativeFeedback nPress={()=>console.log('imagen pr
esionada')}>
      <View style={{
          width:200,
          height:70,
     backgroundColor:'dodgerblue'}}></View>
</TouchableNativeFeedback>

*/
```


```
<Button
  color="orange"
  title="ClickMe" onPress={(console.log("boton clickeado"))}>
</Button>
```


```
<Button
      color="green"
      title="ClickMe" onPress={()=>alert('boton
verde clicked')}>
      </Button>
```

StyleSheet

Este código parece CSS, pero en realidad es JavaScript properties

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#fff",
    justifyContent: "center",
    alignItems: "center",
  },
});
```

Plain java object

```
return (
  <View style={styles.container}>
    <Text>Hola Mundo! </Text>
    <StatusBar style="auto" />
  </View>
);
}
```

Sinlge style

```
return (
  <View style={{backgroundColor: 'orange'}}>
    <Text>Hola Mundo! </Text>
    <StatusBar style="auto" />
  </View>
);
}
```

Definición y Referencia a un objeto

```
return (
  <View style={containerStyle}>
    <Text>Hola Mundo! </Text>
    <StatusBar style="auto" />
  </View>
);
}
const containerStyle = { backgroundColor: 'orange' }
```

Array de estilos para combinarlos

```
return (
  <View style={[styles.container, containerStyle]}>
    <Text>Hola Mundo! </Text>
    <StatusBar style="auto" />
  </View>
);
}
const containerStyle = { backgroundColor: 'orange' }

const styles = StyleSheet.create({
  container: {
```

Platform-specific Code

```jsx
import React from 'react';
import { StyleSheet, Text, View, Platform } from 'react-native';
// detectar plataforma del mobil
export default function App() {
  return (
    <View style={styles.container}>···
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    paddingTop: Platform.OS === "android" ? 20 : 0
  },
});
```

```jsx
import { StyleSheet, Text, View, Platform, StatusBar } from 'react-native';
// detectar plataforma del mobil
export default function App() {
  return (
    <View style={styles.container}>···
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    paddingTop: Platform.OS === StatusBar.currentHeight ? 20 : 0
  },
```

Dimensions

- **D**ensity-**i**ndependent **P**ixels
- Pixels físicos = DIPs * Factor de Escala

Model	4, 4s	11 Pro Max
Points	320 x 480	414 x 896
Scale Factor	2x	3x
Pixels	640 x 960	1242 x 2688
View Width	150 x 2 = 300	150 x 3 = 450

```
return (
  <View style={{
      backgroundColor: 'dodgerblue',
      width: 150,
      height: 70
  }}>
    <Text>Hola Mundo! </Text>
  </View>
```

Exactamente la mitad…. Usamos el %

```
return (
  <View style={{
      backgroundColor: 'dodgerblue',
      width: '50%',
      height: 70
  }}>
    <Text>Hola Mundo! </Text>
  </View>
);
```

Dimensions

```
import { StyleSheet, Text, View, Dimensions } from 'react-native';

export default function App() {
  console.log(Dimensions.get())
  return (
    <View style={{
```

(method) Dimensions.get(dim: "window" | "screen"): ScaledSize

```
Object {
  "fontScale": 1,
  "height": 896,
  "scale": 3,
  "width": 414,
}
```

En IOS (window/screen) son del mismo tamaño
En Android, window es mas pequeña que screen

Dimensions → NO DETECTA la orientación del celular

Detecting Orientation Changes

App.json

app.json > ...

```
{
  "expo": {
    "name": "DoneWithIt",
    "slug": "DoneWithIt",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "splash": {
```

```
"expo": {
  "name": "DoneWithIt",
  "slug": "DoneWithIt",
  "version": "1.0.0",
  "orientation": "landscape",
  "icon": "./assets/icon.png",
```

app.json > {} expo

```
{
  "expo": {
    "name": "DoneWithIt",
    "slug": "DoneWithIt",
    "version": "1.0.0",
    "orientation": "default",
    "icon": "./assets/icon.png",
```

Soporta ambos sentidos de la orientación

Instalar

npm i @react-native-community/hooks

Navigation

Navegación entre Screens

```
C:\Ale\React\PROYECTOS_MOBILE>expo init ejemplos
? Choose a template: » - Use arrow-keys. Return to submit.
    ----- Managed workflow -----
    blank                 a minimal app as clean as an empty canvas
    blank (TypeScript)    same as blank but with TypeScript configuration
>   tabs (TypeScript)     several example screens and tabs using react-navigation and TypeScript
    ----- Bare workflow -----
    minimal               bare and minimal, just the essentials to get you started
    minimal (TypeScript)  same as minimal but with TypeScript configuration
```

- **createStackNavigator**()

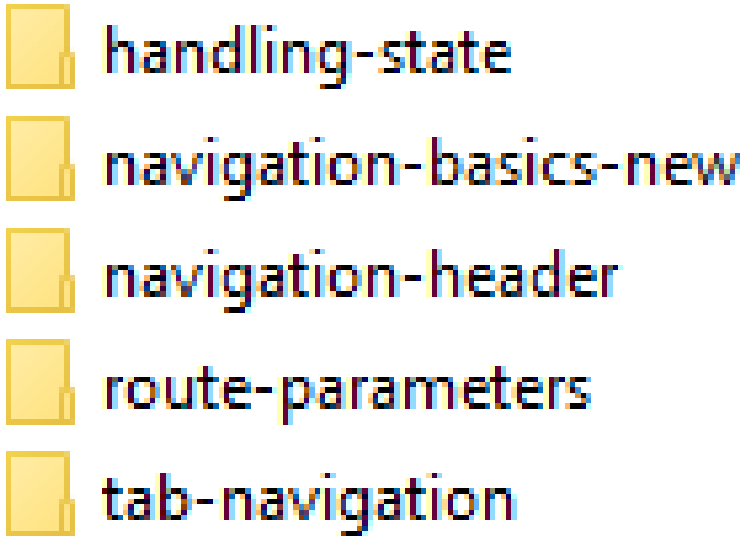
  Hace el setup de la navegacion

Argumentos que recibe

- Home , Settings
- initialRouteName

npm i react-navigation

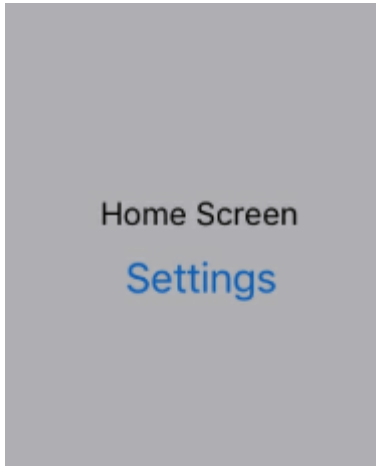
npm i react-navigation-stack



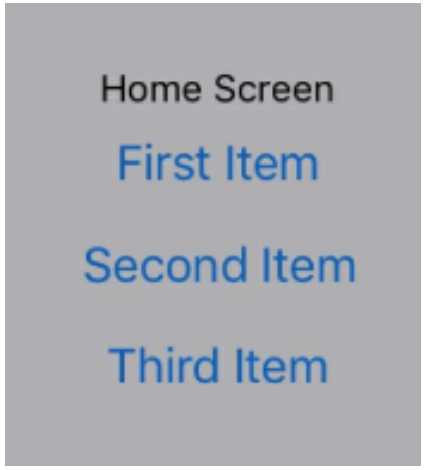
Navegación entre Screens

npm i react-navigation-header-buttons
npm i react-navigation-drawer
npm i react-navigation-tabs
npm i react-native-reanimated

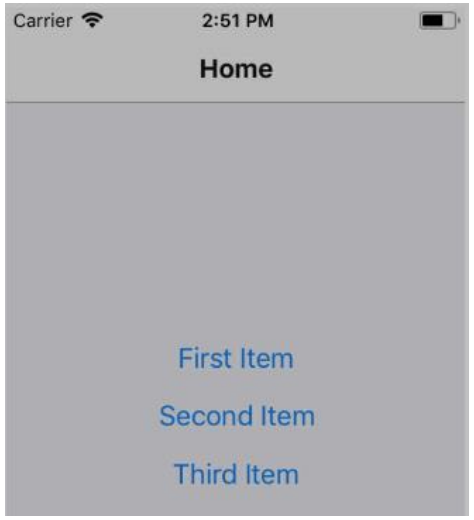
navigation-basics-new

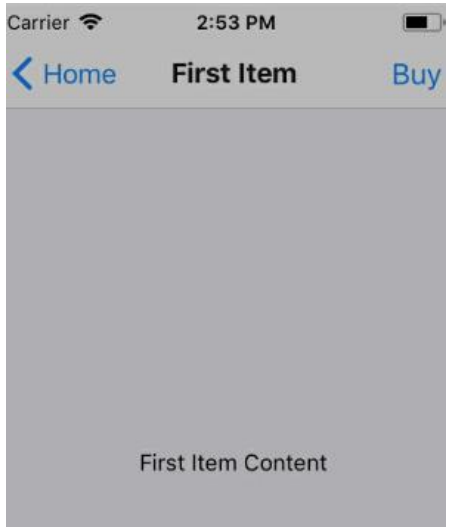


route-parameters

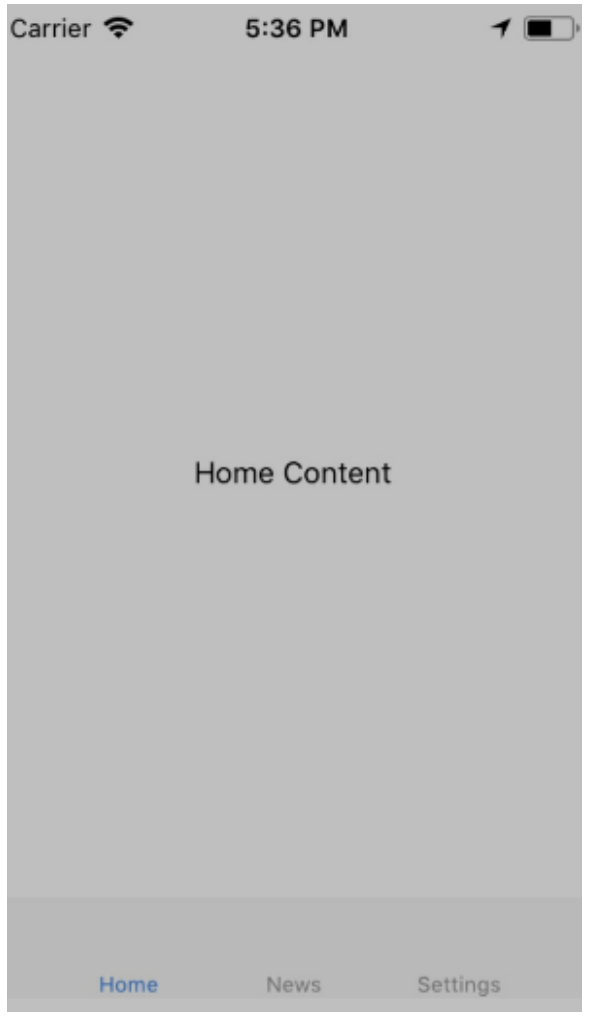


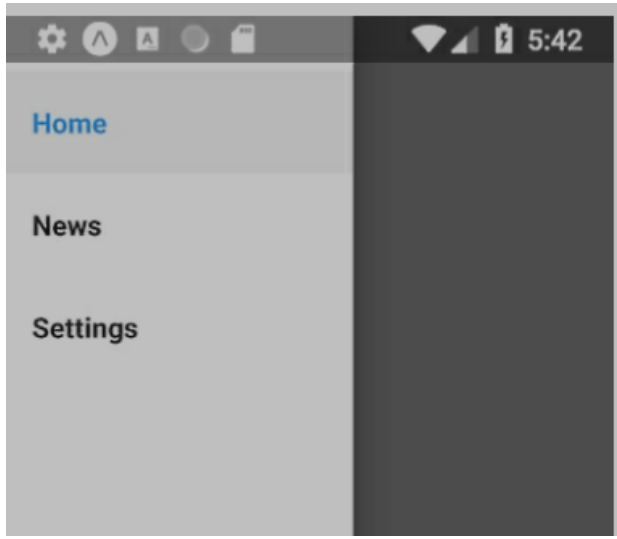
navigation-header





tab-navigation



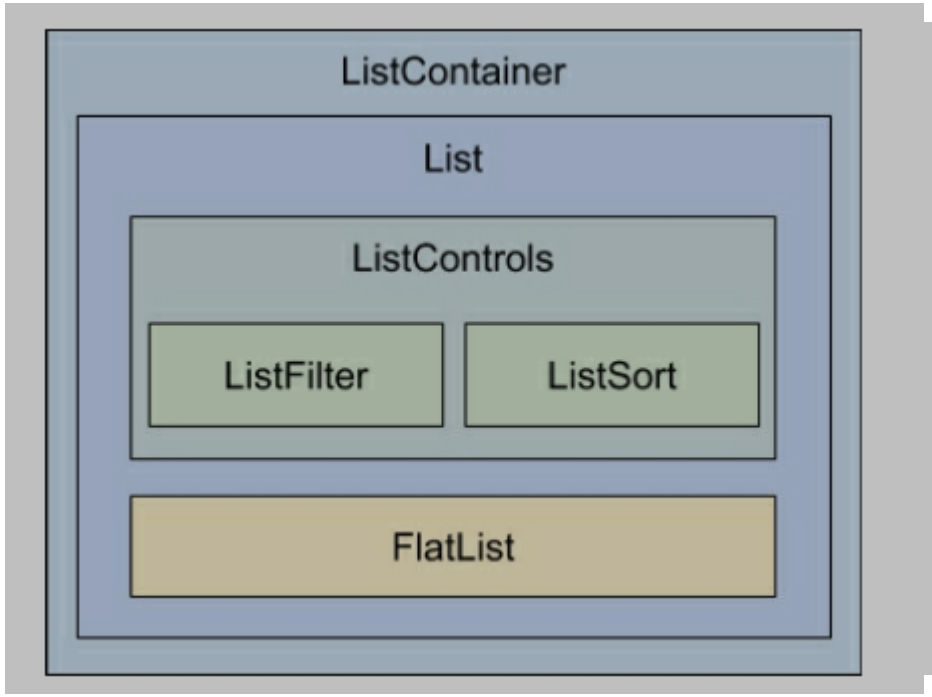


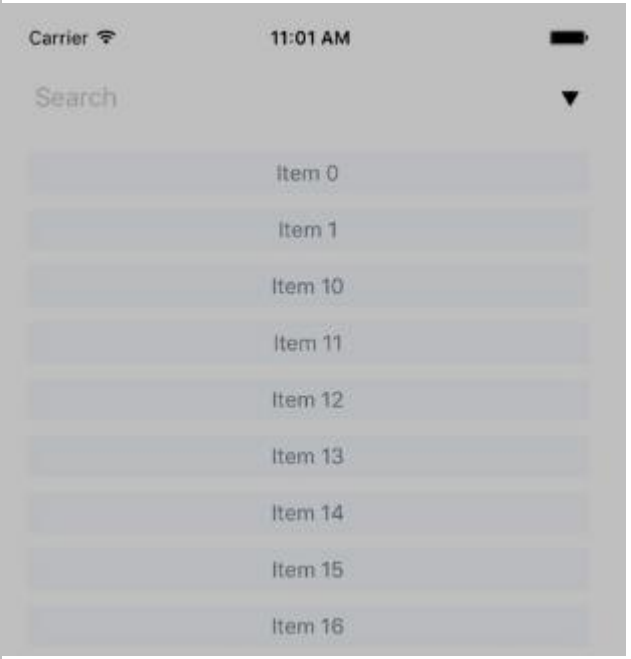
Rendering Item Lists

- fetching-list-data
- lazy-list-loading
- rendering-data-collections
- sorting-and-filtering-lists

rendering-data-collections

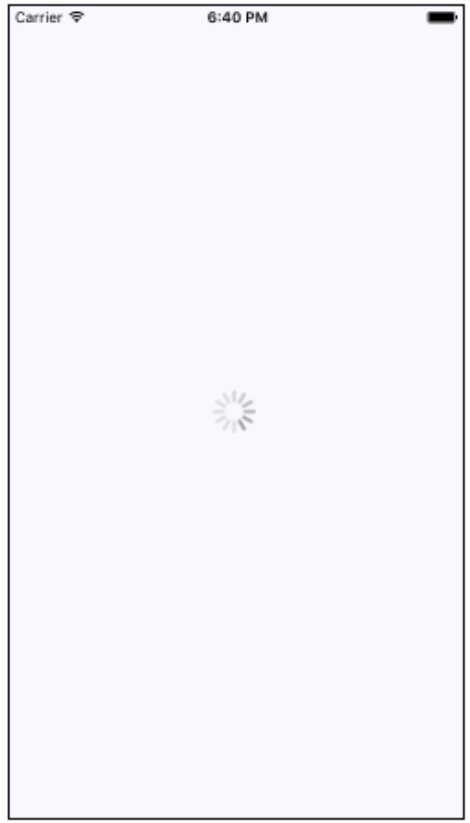
sorting-and-filtering-lists

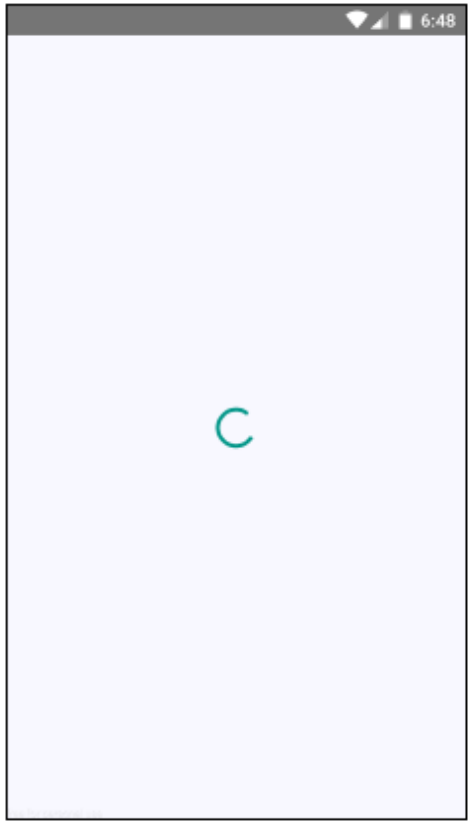


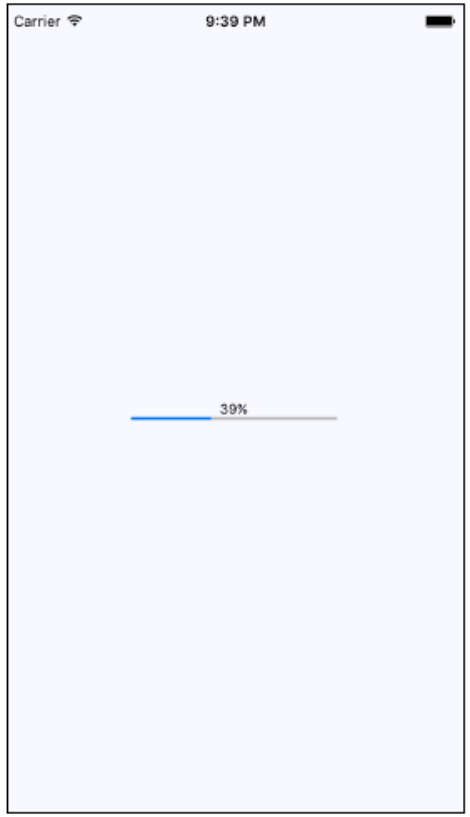


Indicadores de Progreso

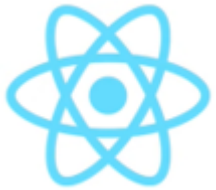
- indicating-progress
- measuring-progress
- navigation-indicators
- step-progress-new







3era CLASE



Geolocalizacion

Mas de Imágenes

User Input

Mas Elementos de Interfaz

- Geolocalizacion and Mapas
  - Donde estoy?
  - Que hay a mi alrededor?
  - Anotacion de puntos de interes
  - Ploting de un Area
- Obteniendo data de User Input
  - Text Input
  - List
  - Check
  - Input de Date/Time
- Controlando Images
  - Loading Images
  - Resizing images
  - Lazy Image loading
  - Rendering Icons
- Respondiendo a Gestos
  - Scrolling con dedos
  - Touch feedback
  - Deslizar

Geolocalizacion

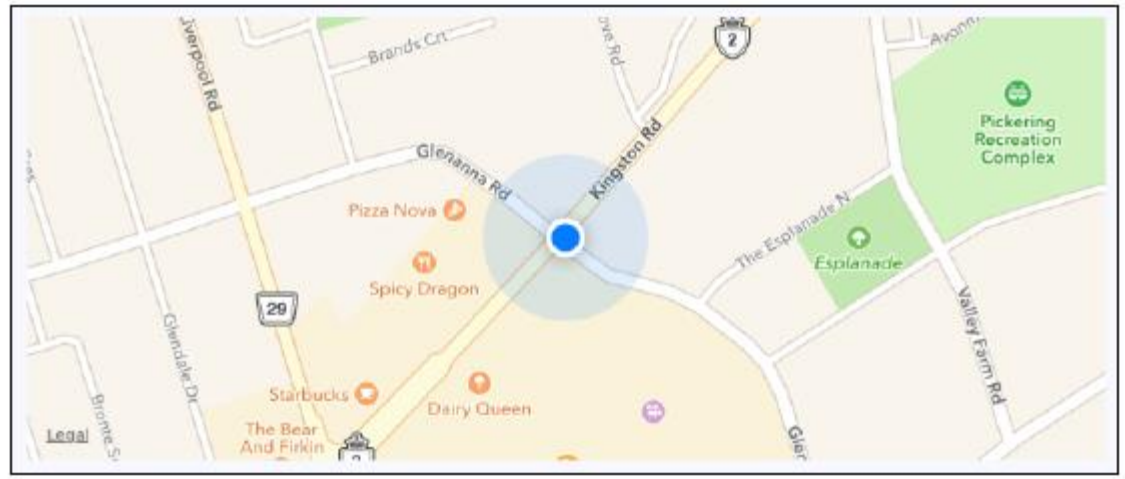
npm i react-native-progress-steps

npm install @react-native-community/progress-bar-android –save
npm install react-native-map --save-exact
npm install --save react-native-open-maps

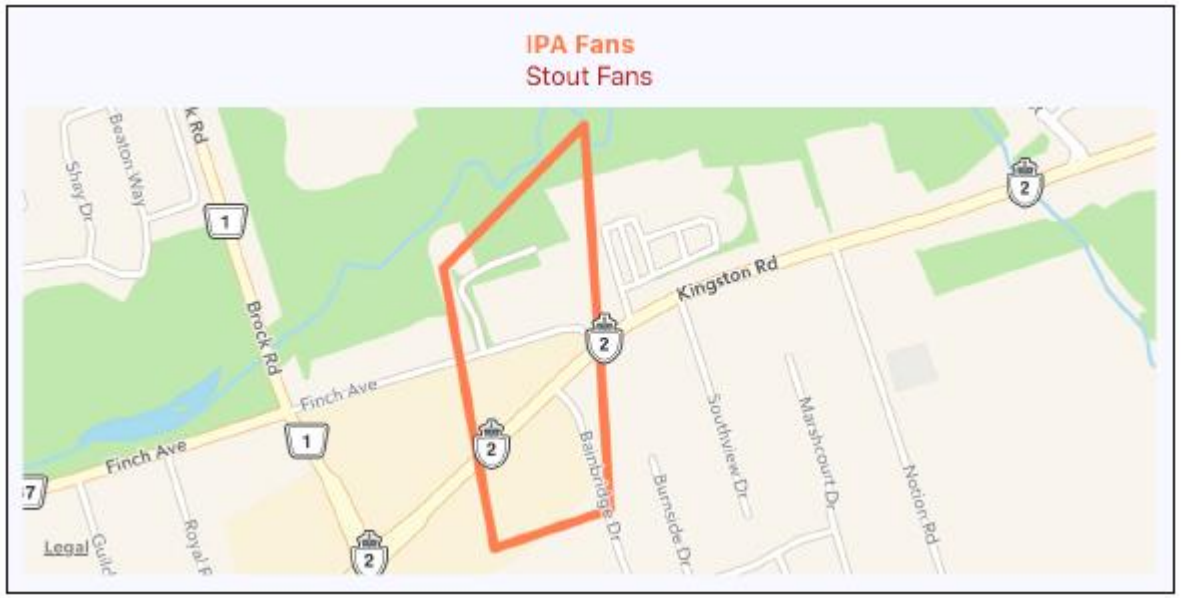
Controlling images
npm install @react-native-community/slider –save
npm install @react-native-picker/picker –save
npm install --save @expo/vector-icons

Mapas
https://blog.logrocket.com/introduction-to-react-native-maps/
Crear una clave para el api de google maps
https://developers.google.com/maps/documentation/android-sdk/get-api-key


Carrier
8:43 AM
Accuracy: 50
Address: 2 Infinite Loop, Cupertino, CA 95014, USA
Altitude: 0
AltitudeAccuracy: -1
Heading: -1
Latitude: 37.33233141
Longitude: -122.0312186
Speed: -1


Liverpool Rd
Brands Crt
Glenanna Rd
Kingston Rd
Pizza Nova
Spicy Dragon
29
Glendale Dr
Starbucks
Dairy Queen
The Bear And Firkin
Legal
Pickering Recreation Complex
The Esplanade N
Esplanade
Valley Farm Rd

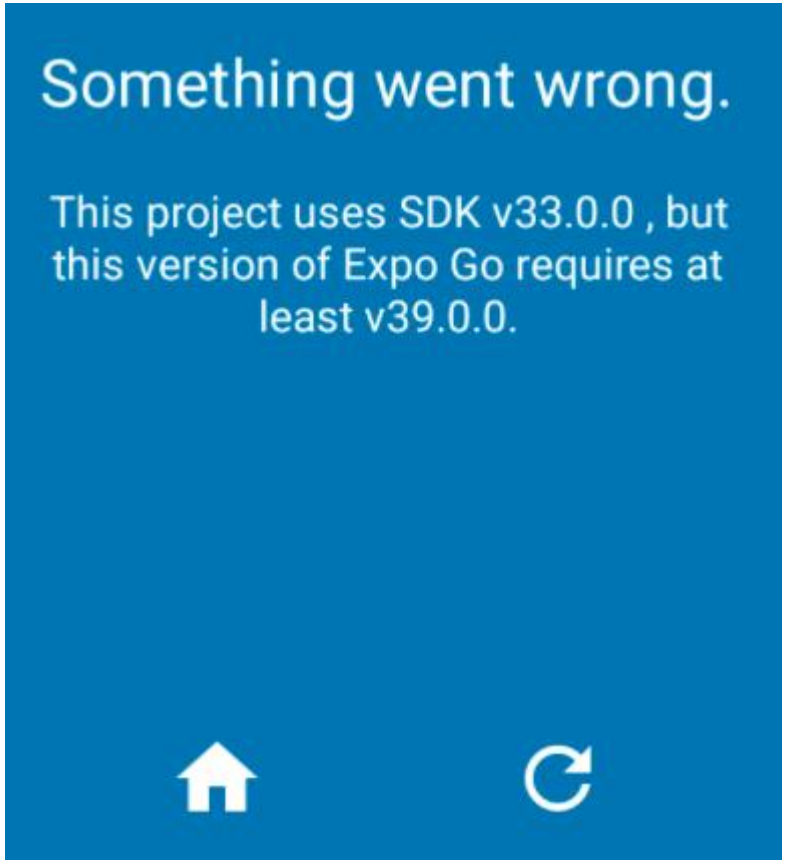

Duff Brewery
Duff beer for me, Duff beer for you
Liverpool Rd
Anton Sq
Glendale Dr
29
Faylee Cres
Fieldlight Blvd
Brands Crt
Glenanna Rd
Glengrove Rd
Glenview Rd
Kingston Rd
Avonmore Sq
Valley Farm Rd
Malden Cres
Legal


IPA Fans
Stout Fans
Beaton Way
Shay Dr
Brock Rd
Kingston Rd
Finch Ave
Finch Ave
Bainbridge Dr
Burnside Dr
Southview Dr
Marshcourt Dr
Notion Rd
Legal


IPA Fans
Stout Fans
Beaton Way
Shay Dr
Brock Rd
Kingston Rd
Finch Ave
Finch Ave
Bainbridge Dr
Burnside Dr
Southview Dr
Marshcourt Dr
Notion Rd
Legal

Tips ante errores de versión

- Upgrading Expo SDK - Expo Documentation
    - https://docs.expo.dev/workflow/upgrading-expo-sdk-walkthrough/#sdk-39
- **expo update 39.0.0**
  …. tener el celu conectado



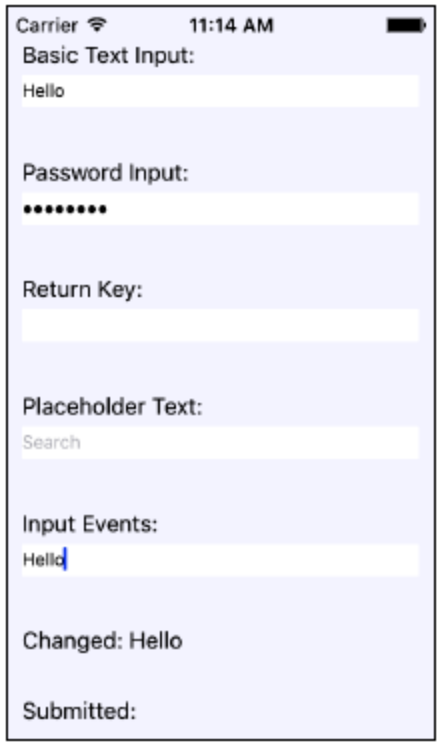
Recolentando Input de Usuario

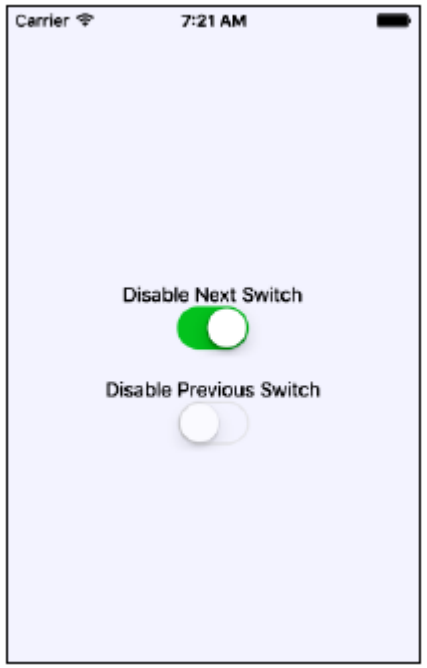
collecting-date-time-input

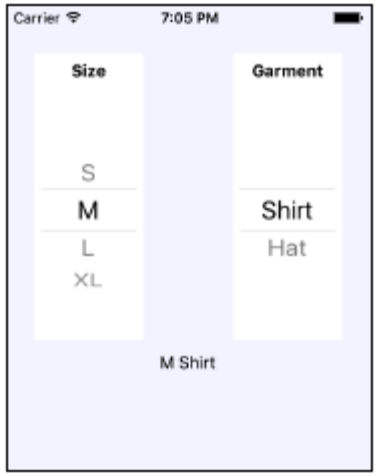
collecting-text-input-new

selecting-options

toggling-on-and-off







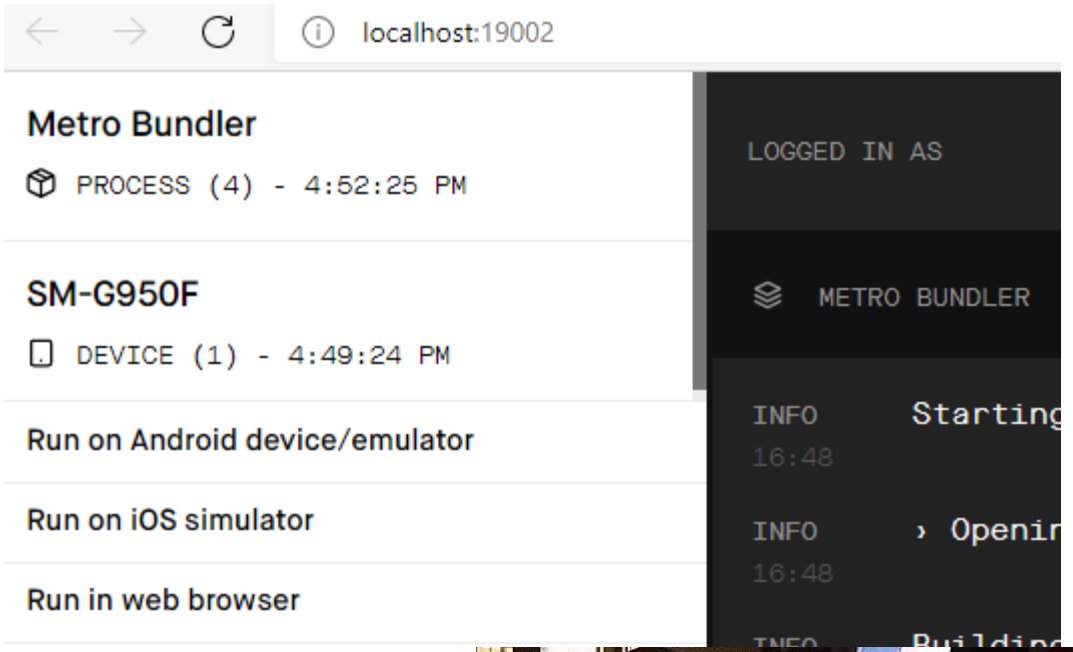
Kick-Starting React Native Projects

en una ventada CMD, ejecutar

1. Chequear la versión de NODE.JS …. Superior a 12
2. **npm install –g expo-cli**
3. expo init **my-project**
4. elegir Managed workflow …… es el default
5. tipear el nombre para la app….. **my-project**

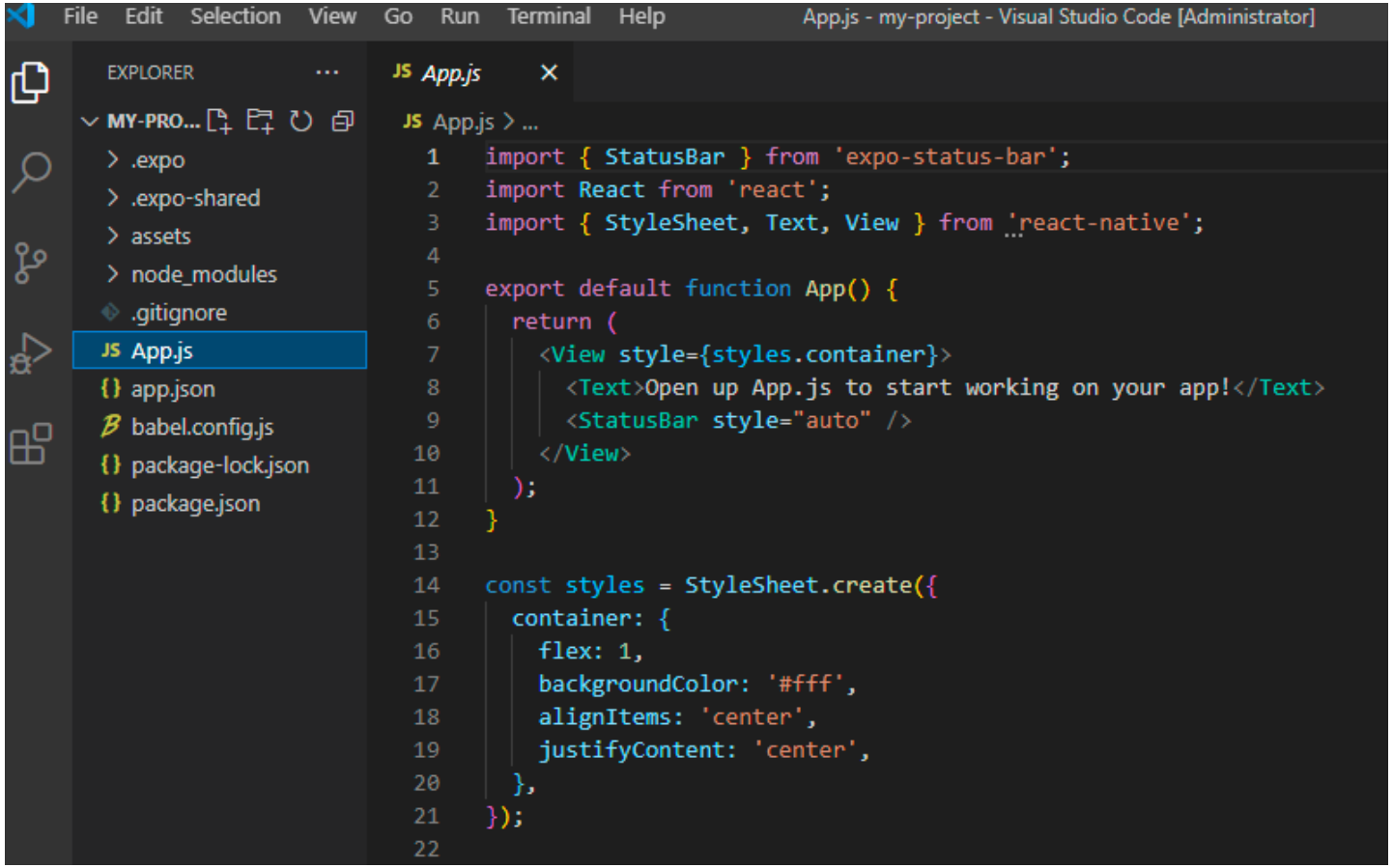
Comprobar la app en el celular

1. En una ventana CMD, ir al directorio del proyecto cd …./**my-project**
2. **npm start**
3. ir a un navegador …. localhost:19002
4. Conectar el celular a la compu
5. Elegir en Connection … LOCAL
6. Click en Run on Android device



Kick-Starting React Native Projects

- Abrir el proyecto en Visual Studio



React Native Bootstrap Styles

- Install

  npm i –g **react-native-bootstrap-styles**

- **Import**

```
import BootstrapStyleSheet from 'react-native-
bootstrap-styles';
```

- npm install -g create-react-native-app
- create-react-native-app HelloWorld
- npn run android