React-Native Fundamentals

(Mobil development)

What is React-Native?

- Is a JavaScript framework to create hybrid apps in Android or IOS platforms, using React library.
- The apps code is written with JavaScript and view interfaces apply HTML fundamentals and CSS.
- Applies the DOM concepts of web pages development.
- React library was written by Facebook develop teams.

Mobile platform development alternatives:

Alternative	Description	Advantages	Disadvantages
Native Code	Java, Kotlin for Android Swift, Objective-C for IOS	Official support Better performance More Flexibility Updated	Only for the specific platform Cost higher Teams with a lot members Inconsistencies between Operative Systems versions
Multiplatform	React Native Flutter Xamarin	Compatibility between Android/IOS. Short time to develop required Less resources required too Uses programming languages known	Less performance that above option Partial support APIs not updated More work to obtains the native sensations of each platform
Web Responsive	HTML + CSS + JS	Compatibility between Operative Systems Hight availability Immediate updates Developer community more extensive	Poor performance, compared that above options Access to native functions is restricted Not appears in the apps store The users not view it as real apps

Install tools and work environment:

Requirements on Windows

- Java Development Kit (in my case have 17.0.3 version) prefers use LTS version

- Android Platform
  - SDK (System Development Kit (Android versions))
  - AVDs (Android Virtual Devices)
    - AVD with same API Android cellphone
  - IDE (Integration Development Environment)

- NodeJS, I uses 24.12.0 LTS version with npm 11.7.0 and yarn 1.22.19

- Cellphone with USB connected (Android 13 or >) + (developer mode must be required)

- Environment variables configuration (both user and system)(change to real address)
  - JAVA_HOME : C:\Program Files\Java\jdk-17.0.3
  - ANDROID_HOME : D:\Programs\Android\SDKs
  - PATH : %ANDORID_HOME%\platform-tools
  - PATH : %JAVA_HOME%\bin

Install tools and work environment:

Requirements on Windows

- Visual Code with the following plugin installed:
  - React Native Tools (v1.13.0) Microsoft.com
  - React Native/React/Redux snippets for es6/es7 (v2.0.6) EQuimper
  - Babel JavaScript (0.0.40) Michael McDermott
  - ES7+ React/Redux/React-Native snippets (v4.4.3) dsznajder

- Create work folder to storage future projects.

Reboot your computer, for changes take effect.

Execute the npm install [<library> or <libraries>] into the project folder, to install the required dependencies, this command will be executed for each project that you create.

Project with Expo-CLI environment:

Expo is a friendly platform and a framework to create universal React applications, with a set of tools and services-built around React-Native platform using JavaScript or TypeScript codebase to develop, debugging, iterate and deploy Web, Android and IOS applications.

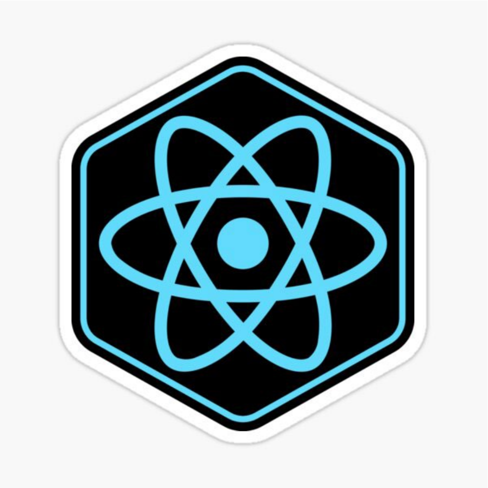
Expo’s qualities allow to speed up development times and reduce development costs.

Expo don’t support handling Objective-C or Kotlin languages, only JS and TS, it’s oriented to beginners' developers.

Expo use many JS libraries to create the applications, which makes then slower and heavier that a native application running on the physical device.

Not all Expo modules are compatible with React Native as default.

Project with React-Native-CLI environment:



Ideal to integrate the native code written in Java or Kotlin in Android or well as Swift or Objetive-C in your IOS project.

It allows the components encapsulation, managing their own states and interactions with other components to create complex applications.

Developers also require knowledge of native languages to achieve a similar feel to Android an IOS applications.

Applications created with this library tend to be heavier and slower than native Android and IOS applications.

Creating projects:

Step 01: Open Power Shell or command prompt terminal and change directory to React main folder to storage your homework's.

Step 02: Into the terminal window, execute the next command to create a new React-Native project:

```
npx create-expo-app@latest

    ? What is your app named?.....:  demo_01
```

Step 03: For testing your code, execute the next steps:

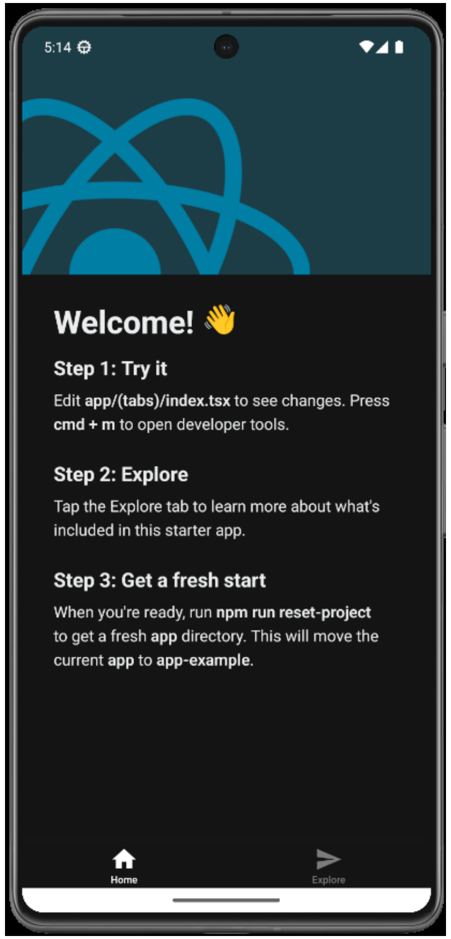
- Connect your physical cellphone with USB cable (optional) (Expo Go app is required).
- Open the Android Studio and run the AVD target. (required).

Step 04: Change directory to new folder project **demo_01**.

Step 05: Execute the next command:

```
- npm run android
- npm run ios  # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

You will appreciate in the AVD or physical cellphone the running application.



Welcome! 👋

Step 1: Try it

Edit **app/(tabs)/index.tsx** to see changes. Press **cmd + m** to open developer tools.

Step 2: Explore

Tap the Explore tab to learn more about what's included in this starter app.

Step 3: Get a fresh start

When you're ready, run **npm run reset-project** to get a fresh **app** directory. This will move the current **app** to **app-example**.