

Native and Non-Native Apps

Islas Luis

January de 2024

Native Apps

Native apps are tightly integrated with the operating system of the mobile devices for which they were designed. For example, native iOS apps are written in Swift or Objective-C, while Android apps are developed in Kotlin or Java. These languages are optimized to take full advantage of platform-specific features and controls. During compilation, the source code is translated directly into binary code, resulting in optimal performance and direct access to all hardware and operating system features. Native apps are built to run directly in the app. When you build a native mobile app, the source code (human-readable) is actually converted to binary (machine-readable) code and runs from there.

Examples of native apps include:

- **WhatsApp (iOS y Android):** WhatsApp is an instant messaging app that takes full advantage of platform-specific features. On iOS devices, it's written in Objective-C or Swift, while on Android it's Java or Kotlin. This allows for deep integration with each platform's operating system capabilities.
- **Instagram (iOS y Android):** The Instagram app, used to share photos and videos, is developed natively for iOS in Swift or Objective-C and for Android in Java or Kotlin. This ensures optimal performance and a consistent user experience across each platform.
- **Camera+ (iOS):** This photography app for iOS devices is specifically designed to take full advantage of the camera capabilities and editing features available on Apple devices. It's written in Swift or Objective-C.
- **Google Maps (iOS y Android):** Although Google Maps is available on both platforms, each version is developed natively. The iOS app is written in Swift or Objective-C, while the Android app uses Java or Kotlin. This allows for full integration with each platform's navigation and mapping features.

Non-native apps

Non-native mobile apps are developed in high-level languages, such as JavaScript, CSS, HTML5, Python, or .NET, and cannot directly access operating system and hardware features. To overcome this limitation, they require an additional layer or framework that acts as an intermediary to use the specific operating system controls designed by Google and Apple for Android and iOS, respectively.

For non-native applications, the source code is not compiled directly into binary code. Instead, the source code is stored internally in the application. When the app starts, it reads the source code from stored files and executes the code from there. It's important to note that this approach implies that the source code is available to anyone who has access to the app's files, and these files are not encrypted by default, which could raise security concerns. The topic of security will be specifically addressed in a future blog post related to this same topic.

Examples of non-native apps include:

- **Facebook (React Native):** Facebook uses React Native to develop its core app. React Native is a development framework that allows you to write code in JavaScript and React to build apps that work on multiple platforms, such as iOS and Android.
- **Airbnb (React Native):** Similar to Facebook, Airbnb uses React Native to develop its app. This allows them to share a large portion of the code between iOS and Android versions, simplifying the development and maintenance process.
- **Twitter (Flutter):** Twitter has adopted Flutter, a Google development framework, to build its app. Flutter enables cross-platform application development using the Dart programming language.
- **LinkedIn (Xamarin):** LinkedIn uses Xamarin, a development platform that allows you to write apps in C Sharp to work on iOS, Android, and Windows. This makes it easy to develop cross-platform without having to maintain different codebases.