TECHNOLOGICAL UNIVERSITY OF TIJUANA

App designs

4 D

Native, non-native and cross-platform applications

Gutierrez Palomares Luis Alberto

January 10, 2025

DR. Ray Brunett Parra Galaviz

Native, non-native and cross-platform applications

Mobile applications are essential to improve the experience of users using the device, but they have ways of being developed depending on the needs of the project and the limitations of the device. The concepts of native, non-native and cross-platform applications will be described below.

Native Applications

Native apps are developed specifically for a particular operating system, such as Android or iOS. They use programming languages and tools provided by the operating system manufacturers.

Technical Details

Android:

- Languages: Java, Kotlin.
- Recommended IDE: Android Studio.
- APIs: Android SDK.

iOS:

- Languages: Swift, Objective-C.
- Recommended IDE: Xcode.
- APIs: iOS SDK.

Operation

These apps run directly on the device's hardware, ensuring maximum performance and full access to operating system features such as cameras, sensors, storage, etc.

Use Cases

Applications that require high performance (games, intensive graphics tools).

Apps with deep integration with hardware (GPS, advanced cameras, biometric sensors).

Personalized and optimized user experiences.

Non-Native Applications

Non-native applications are accessible through web browsers. They are not installed on the device and are mostly designed to be compatible with any device with a browser.

Technical Details

Languages: HTML, CSS, JavaScript.

Frameworks and tools: React.js, Angular, Vue.js.

Advanced implementations: PWAs (Progressive Web Apps), which allow functionalities such as push notifications, offline storage and the appearance of native applications.

Operation

They work within a browser and depend on an Internet connection (although PWAs can offer offline functionality).

They require no installation, saving space on your device.

Use Cases

Services that need to be accessible from multiple platforms without installation.

Lightweight applications or prototypes that do not justify native development.

E-commerce and management systems (for example, ERP systems accessible from any device).

Cross-platform applications

Cross-platform apps combine the best of both worlds, allowing a single code base to work across multiple operating systems (Android, iOS, even web).

Technical Details

Popular frameworks and languages:

- React Native: JavaScript.
- Flutter: Dart.
- Xamarin: C#.
- Ionic: HTML, CSS, JavaScript.
- Unity: For cross-platform games.

Strategies:

- Compilation to native code: Some tools convert the code to an optimized native format.
- Hybrid rendering: Use a web container to render the application within a native framework.

Operation

They offer a balance between performance and cross-platform reach.

Many tools allow access to native APIs through plugins or modules.

Use Cases

Projects with budget or time limitations, where the aim is to launch a functional app on several platforms quickly.

Applications that do not require extremely high performance.

Startups that need to validate a product before investing in native development.