



Program. Multiplatform Software Development

Group. 4°D

Student. Lara López Josue Isaac

Subject. App Design

Activity. Native, non-native and cross-platform applications

Professor. Ray Brunett Parra Galaviz

Tijuana, B.C., January 10, 2024.

NATIVE, NON-NATIVE AND CROSS-PLATFORM APPLICATIONS.

Native, non-native, and cross-platform applications represent different approaches to mobile app development, each with distinct characteristics, advantages, and challenges.

Native Applications:

Native apps are developed specifically for a single platform, such as iOS or Android, using platform-specific programming languages and tools.

Advantages:

- **Performance:** Optimized for the target platform, resulting in faster and more responsive applications.
- **User Experience:** Seamless integration with the platform's UI/UX conventions, providing a familiar look and feel to users.
- **Access to Device Features:** Full access to all native APIs and device hardware capabilities.

Disadvantages:

- **Development Cost:** Separate codebases are required for each platform, leading to higher development and maintenance costs.
- **Time-Consuming:** Developing and updating multiple versions of the app can be time-intensive.

Non-Native Applications:

Non-native apps, often referred to as web or hybrid apps, are built using web technologies like HTML, CSS, and JavaScript and run within a web browser or a webview component in a native shell.

Advantages:

- **Cross-Platform Compatibility:** A single codebase can run on multiple platforms, reducing development effort.
- **Cost-Effective:** Lower development and maintenance costs due to shared code.

Disadvantages:

- **Performance Limitations:** Generally slower and less responsive compared to native apps.

- **Limited Access to Device Features:** Restricted access to certain device functionalities and native APIs.

Cross-Platform Applications:

Cross-platform apps are developed using frameworks that allow a single codebase to be compiled into native code for multiple platforms. Popular frameworks include React Native, Flutter, and Xamarin.

Advantages:

- **Code Reusability:** Significant portions of code can be shared across platforms, streamlining development.
- **Cost and Time Efficiency:** Reduced development time and costs compared to building separate native apps.

Disadvantages:

- **Performance Variability:** May not match the performance of fully native apps, though often sufficient for many use cases.
- **Platform Limitations:** Some native features may be harder to implement or may require platform-specific code.

Choosing the Right Approach:

The decision between native, non-native, and cross-platform development depends on factors such as project requirements, budget, timeline, and desired user experience. Native development is ideal for performance-critical applications requiring deep integration with device features. Cross-platform development offers a balance between performance and development efficiency, suitable for many business applications. Non-native (web or hybrid) apps can be appropriate for content-focused applications with limited need for native device interactions.