

# Native and Non Native Applications

Arce Llamas Irvin de Jesus  
Multiplatform Software Development  
Universidad Tecnológica de Tijuana  
4A 06/02/24

## Native Apps

**Definition:** Native apps are specifically designed for a particular operating system (OS), such as iOS (Apple) or Android (Google). Here are some key points:

- **Performance:** Native apps are optimized for the specific OS and hardware, resulting in better performance.
- **User Experience:** They provide a consistent and seamless experience, adhering to the OS's design guidelines.
- **Access to Hardware:** Native apps can directly utilize device features like the camera, GPS, and sensors.

### Development Languages:

- **Android:** Native Android apps are typically written in Java or Kotlin.
- **iOS:** Native iOS apps are written in either Swift or Objective-C.

**Example:** A native Android app can directly leverage Android-specific features and hardware.

## Non-Native Apps

**Definition:** Non-native apps are not tied to a specific OS and can run on multiple platforms. They include various sub-categories:

- **Hybrid Apps:** These blend elements of both native and web apps. They use web technologies (like HTML, CSS, and JavaScript) within a native container.
- **Cross-Platform Apps:** These allow developers to write code once and deploy it across different platforms (e.g., iOS, Android, and even web).
- **Progressive Web Apps (PWAs):** These are web applications that offer an app-like experience when accessed through a browser.

### Advantages:

- **Cost-Effective:** Non-native apps can save development costs by sharing code across platforms.
- **Faster Development:** They can be developed more quickly than native apps.
- **Platform Independence:** They work across different OSes.

**Example:** A cross-platform app built using a framework like React Native or Flutter.