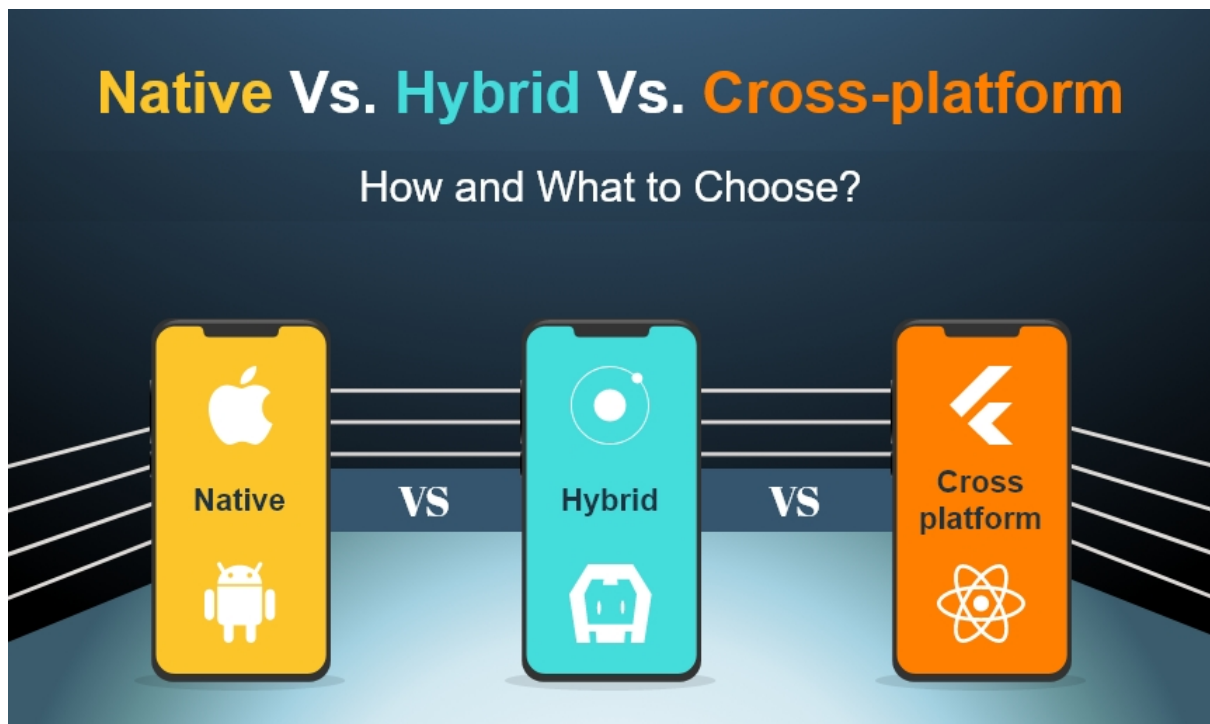


Universidad Tecnológica de Tijuana



Alumno: Alcantara Huerta Angel Josue

Docente: Ray Brunett Parra Galaviz.

Materia: 4D-Diseño de APPS 2025-1.

Trabajo: Native, Hybrid and Multiplatform Application Development

Fecha de entrega: 10 de enero de 2025

Native, Hybrid and Multiplatform Application Development

Mobile application development has evolved to offer a variety of methodologies to suit different business and technical needs. These methodologies include native, hybrid and cross-platform application development, each with its own advantages and disadvantages. These methodologies are explored in detail below to help better understand their implications and applications.

Native Application Development

Native application development involves creating applications specifically for a particular platform, using that platform's native programming languages and tools. For example, developers use Java or Kotlin for Android apps, and Objective-C or Swift for iOS apps.

Advantages:

- **Optimal Performance:** Because applications are designed specifically for a platform, they can take full advantage of the hardware and operating system capabilities, resulting in superior performance.
- **User Experience (UX):** Native applications can offer a highly customized and optimized UX, conforming to the design guidelines of each platform.
- **Full Hardware Access:** Enables direct access to device functions, such as camera, GPS, and sensors, offering deeper and more efficient integration.

Disadvantages:

- **Cost and Time:** Requires more resources and time, as each platform needs its own development and maintenance.
- **Specialized Teams:** Needs developers with specific knowledge in the corresponding platforms.

Hybrid Application Development

Hybrid application development uses web technologies such as HTML, CSS, and JavaScript to create applications that run in a native container. This allows hybrid applications to be deployed on multiple platforms with a single code base.

Advantages:

- **Development Speed:** Allows for faster application creation, ideal for time-constrained projects.
- **Cost Effective:** With a single code base, development and maintenance costs are significantly lower.
- **Multi-Platform Deployment:** Facilitates deployment on multiple platforms simultaneously.

Disadvantages:

- **Performance:** Can be less efficient than native applications due to dependence on an integrated browser.
- **Limited UX:** User interfaces can be inconsistent and less fluid compared to native applications.

Cross-Platform Application Development

This approach allows code to run natively on multiple platforms using a cross-platform development framework such as React Native, Flutter, or Xamarin. These frameworks allow you to “write once, run anywhere”.

Advantages:

- **Cost and Time Efficiency:** Significant savings in development time and costs by reusing code.
- **Improved UX:** Modern frameworks have improved in offering user experiences closer to native.
- **Easy Maintenance:** A single team can maintain the application, simplifying updates and fixes.

Disadvantages:

- **Inferior Performance to Native:** Although improved, performance still may not match native applications, especially on complex tasks.
- **Hardware Access:** May be limited or slower, depending on the framework used.

Considerations for Choosing a Development Approach

When deciding between native, hybrid or cross-platform development, it is critical to consider several factors:

- **Application Complexity:** Highly complex applications often benefit from the native approach.
- **Speed to Market:** If time is of the essence, hybrid or cross-platform approaches may be more appropriate.
- **Budget:** Hybrid and cross-platform approaches are more cost effective.
- **Target Audience:** User preference for a specific platform may influence the choice.
- **Access to Device Features:** Applications that rely on advanced hardware capabilities may require native development.
- **Security:** Native applications typically offer greater built-in security capabilities.