



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Topic:

Framework Selection

By:

Arguelles Galvez Antonio

Group:

10B

Matter:

Integral Mobile Development

Teacher:

Ray Brunett Parra Galaviz

Date:

01/09/2025

Choosing the right framework is a crucial step in software development, as it directly impacts the project's efficiency, scalability, and success. A framework provides a structured foundation with predefined code, tools, and best practices to facilitate development.

Key Factors for Choosing a Framework

- **Project Requirements:** The framework should align with the specific functionalities and goals of the project.
- **Language Compatibility:** It must be compatible with the programming language chosen by the development team.
- **Community and Popularity:** A framework with an active community ensures access to resources, plugins, and constant updates.
- **Learning Curve:** A framework that is easy to learn reduces onboarding time and boosts productivity.
- **Performance and Scalability:** The framework should handle the expected workload and support future growth.
- **Security:** Built-in security features protect both the application and its users from vulnerabilities.
- **Licensing and Costs:** It is important to consider licensing terms and potential costs, especially for commercial use.
- **Integration Capabilities:** The framework should support integration with databases, APIs, and other tools.
- **Maintenance and Updates:** Choosing a framework that is actively maintained ensures stability and longevity.
- **Testing Support:** Built-in testing tools or compatibility with third-party testing frameworks improve code reliability.

What is React Native?

React Native is an open-source framework developed by Meta (formerly Facebook) that allows developers to build mobile applications for iOS and Android using JavaScript and React. It provides a way to create apps that look and feel like native apps without having to write two separate codebases for each platform.

React Native uses native components, meaning that it translates the code into native views, resulting in better performance compared to traditional hybrid apps. It allows developers to reuse a large portion of code across platforms, making it more efficient and reducing development time.

Key Features of React Native

- **Cross-Platform Development:** Write one codebase that runs on both iOS and Android.
- **Hot Reloading:** Enables developers to see changes instantly without rebuilding the entire app.
- **Native Components:** Uses native UI components to improve performance and user experience.
- **Reusable Code:** Around 90% of the code can be shared between platforms.
- **Third-Party Plugins:** Supports integration with third-party libraries and plugins for extended functionality.
- **Strong Community:** React Native has a large and active developer community, providing support, documentation, and plugins.

Advantages of React Native

- **Faster Development:** The ability to share code between iOS and Android significantly speeds up the development process.
- **Cost-Effective:** Since a single codebase is used for both platforms, development and maintenance costs are reduced.
- **Native-Like Performance:** React Native apps provide a performance level comparable to fully native apps.

- **Large Community and Ecosystem:** A wide variety of libraries, tools, and plugins are available to support developers.
- **Hot Reloading:** Boosts productivity by allowing developers to instantly view code changes without restarting the app.
- **Scalability:** React Native apps can handle both small and large-scale projects.

Disadvantages of React Native

Limited Native Functionality: For complex features requiring heavy native functionality, additional bridging may be needed.

Performance: While React Native performs well, fully native apps may have a slight performance edge in highly demanding applications.

Lack of Some Native APIs: Some platform-specific APIs may not be available in React Native and require custom native modules.

Conclusion

React Native is a powerful framework for building cross-platform mobile applications that provide native-like performance using JavaScript and React. It stands out for its ability to reuse code across platforms, its cost-effectiveness, and strong community support. With features like hot reloading and support for third-party plugins, it improves development speed and efficiency. However, for projects requiring heavy native functionality, developers may need to write additional native code.

Choosing React Native is ideal for projects where time-to-market, cost-efficiency, and cross-platform compatibility are priorities.