

Tarea Extraclase #3

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Estructuras de Datos y Análisis de Algoritmos I (CE 1103)

Semestre II, 2018

Profesor: Antonio González Torres

Estudiantes:

- Ricardo Artavia Solano (2013031079)
 - Rodolfo Bonilla Camacho (2017239550)
 - Romario Ramírez Ramírez (2017185307)
-

Objetivos General:

- Investigar sobre herramientas para el desarrollo de aplicaciones mobile.

Objetivos Específicos:

- Investigar y comparar herramientas disponibles para el desarrollo de aplicaciones mobile.

Introducción

Las aplicaciones móviles son programas de software destinados brindar una solución específica al usuario los cuales son implementados para funcionar en dispositivos móviles. De acuerdo con datos estadísticos, para el 2017 habían registradas 8 millones de aplicaciones móviles en Play Store y 2.2 millones en Apple Store. Además, se estima que los usuarios invierten un 50% del tiempo que utilizan para consumir contenidos digitales, lo hacen a través de plataformas móviles. Hoy en día las aplicaciones móviles permiten automatizar y facilitar todos los aspectos de nuestras vidas. Es por estas razones que el desarrollo de habilidades para diseñar y mantener aplicaciones móviles se vuelve esencial para competir en el mercado digital multimedia.

En este documento se pretende realizar una revisión de tres herramientas de vanguardia que permiten desarrollar aplicaciones móviles y posteriormente realizar una prueba que correrá el sistema operativo Android utilizando Android Studio para su desarrollo.

Herramientas para desarrollo móvil

1. Ionic:

Ionic nació en el 2012 y es un ambiente de desarrollo open sour de aplicaciones móviles. Es multiplataforma, está basado en CSS y una de sus características más aprovechadas es que se puede integrar con Angular JS. También tiene librerías de HTML, CSS y JS optimizadas para ambientes “mobile”.

Ventajas:

- Puede imitar el “look and feel” de la plataforma de destino.
- Imita los gestos y comportamiento de la plataforma de destino.
- Se puede utilizar junto con Apache Cordova.
- Todas sus bibliotecas y utilidades son gratis.
- Se puede integrar con Angular JS.

Desventajas:

- En algunas ocasiones el rendimiento se puede ver afectado debido a que no es una herramienta nativa para los sistemas operativos de destino.
- Para aplicaciones que requieran altos niveles de seguridad debido a que manejan datos sensibles, puede representar una vulnerabilidad debido a su característica de código abierto.
- Las aplicaciones navegables pueden ser complejas de trabajar.
- No recomendable para aplicaciones como juegos o similares que requieran poder de procesamiento gráfico.

2. jQuery Mobile:

Es una de las herramientas de desarrollo con más tiempo en el mercado (desde 2010) y se dice que aproximadamente la mitad de los sitios web móviles actuales utilizan este framework. Es multiplataforma e integra HTML5, CSS, jQuery y jQuery UI de manera organizada y robusta. Esta construido sobre jQuery y es sumamente veloz debido a que utiliza dependencias mínimas, principalmente para el manejo de imágenes.

Ventajas:

- Multiplataforma en cuanto a buscadores web y sistemas operativos

- Ofrece transiciones, eventos de mouse y touch.
- Tiene alto grado de personalización mediante la utilización de temas.
- Sencillo de aprender.
- Tiene una excelente documentación y soporte.
- Es más sencillo obtener permisos para comercializar por parte de las tiendas de Apps como Play Store y Apple Store

Desventajas:

- Carece de una UI avanzada.
- No es de código abierto y algunas funcionalidades pueden estar limitadas.
- Las animaciones y transiciones carecen de fluidez.
- La documentación generada no es limpia.
- Se centra en el desarrollo con JavaScript y esto entorpece la utilización de otros lenguajes o herramientas.

3. XCode:

Es el IDE oficial de Apple de desarrollo de aplicaciones para las plataformas que utilizan IOS como sistema operativo. Tiene todas las funcionalidades necesarias integradas o bien se pueden implementar aquellas que se requieran mediante librerías. No es de código abierto y sus versiones están en constante actualización por parte de Apple. Se puede escribir código en lenguajes como Swift, Objective C, C++, AppleScript, Python, Ruby y Java principalmente. Tiene una herramienta de *Layout* que imita el “look and feel” de las plataformas de destino. Fue lanzado en el 2003.

Ventajas:

- Se puede realizar todo el proceso de desarrollo incluyendo la interfaz gráfica.
- La seguridad es mejor debido a que es controlado por una sola fuente.
- Se mantiene en constante actualización
- Apple provee documentación de calidad.
- Tiene excelentes herramientas de testeo.
- Su interfaz es amigable con el usuario y esto facilita su uso y aprendizaje.
- O requiere tantos recursos del procesador como otras herramientas.

Desventajas:

- Está limitado a producir aplicaciones para la familia de productos Apple.

- Debe ser implementado desde una computadora Apple para que su funcionamiento sea óptimo.
- No es de código abierto.
- La comunidad de desarrolladores es pequeña en comparación a la de otras herramientas.
- No se integra bien con servicios almacenamiento web (servers), se deben utilizar otras herramientas.
- Es necesario aprender Swift y Objective C para aprovechar su potencial.

4. Android Studio:

Es el IDE oficial para desarrollo de aplicaciones en Android. Fue creado por Google específicamente para crear aplicaciones móviles del sistema operativo Android. Basa su funcionamiento en IntelliJ IDEA y ofrece varias herramientas como un emulador, compilador flexible basado en Gradle, instant Run para aplicar cambios sin necesidad de recompilar un nuevo APK, compatibilidad con C++ e integración con Google Cloud Platform. Su estructura funciona mediante módulos: apps, bibliotecas y Google App Engine.

Ventajas:

- Potente emulador flexible basado en Gradle
- Desarrollo unificado para todos los dispositivos Android desde la versión 4.4.
- Plantillas de código e integración con GitHub.
- Gran cantidad de frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad y compatibilidad.
- Integración con XML y Java.
- Nativo para Android por lo tanto se pueden aprovechar al máximo las capacidades de la plataforma de destino.

Desventajas:

- No es multiplataforma.
- No es open source.
- Utiliza gran cantidad de recursos computacionales para su ejecución.
- Algunos sistemas de construcción pueden resultar complejos de aprender al inicio.
- Tiene menor número de plug-ins en comparación con otros ambientes de desarrollo.

