

INVESTIGACIÓN Y PRUEBA DE CONCEPTOS

DESARROLLO DE APLICACIONES EMPRESARIALES

Profesor:

Felipe Botero

HERNÁN DARÍO VELÁSQUEZ ORTIZ

GIOVANI CARDONA MARIN

Universidad de Antioquia

2016

Contenido

| | |
|---|----|
| 1. INTRODUCCION | 4 |
| 2. Aplicaciones web. | 4 |
| 2.1 Ventajas de las aplicaciones web | 5 |
| 2.2 Desventajas de las aplicaciones web..... | 5 |
| 3. Aplicaciones nativas. | 5 |
| 3.1 Ventajas de las aplicaciones nativas..... | 6 |
| 3.2 Desventajas de las aplicaciones nativas | 6 |
| 4. Aplicaciones híbridas | 7 |
| 4.1 Ventajas de una aplicación Híbrida | 7 |
| 4.2 Desventajas de una aplicación Híbrida | 7 |
| 5. PhoneGap..... | 8 |
| 5.1 Ventajas | 9 |
| 5.2 Desventajas | 9 |
| 5.3 Licencia | 10 |
| 6. Fabric..... | 10 |
| 6.1 Ventajas | 10 |
| 6.2 Desventajas | 11 |
| 6.3 Licencia | 11 |
| 7. Ionic..... | 11 |
| 7.1 Ventajas | 11 |
| 7.2 Desventajas | 12 |
| 7.3 Licencia | 12 |
| 8. App Inventor. | 12 |
| 8.1 Ventajas | 13 |
| 8.2 Desventajas | 13 |
| 8.3 Licencia | 13 |
| 9. Android Studio | 14 |
| 9.1 Ventajas | 15 |
| 9.2 Desventajas | 15 |
| 9.3 Licencia | 15 |
| 10. XCode | 15 |
| 10.1 Ventajas | 16 |
| 10.2 Desventajas | 16 |
| 10.3 Licencia..... | 16 |

| | | |
|------|-----------------------------------|----|
| 11. | Características a evaluar | 16 |
| 11.1 | Rendimiento..... | 17 |
| 11.2 | Curva de aprendizaje | 17 |
| 11.3 | Conocimientos actuales | 17 |
| 11.4 | Costo de licencias..... | 17 |
| 12. | Comparación entre frameworks..... | 18 |
| 13. | Selección de framework | 18 |
| 14. | Prueba de conceptos..... | 18 |
| 15. | Bibliografía..... | 20 |

1. INTRODUCCION

Para este trabajo comparativo vamos a investigar y comparar algunos framework de desarrollo de aplicaciones para dispositivos móviles y las herramientas nativas para la creación de estos mismos, así dicho, evaluaremos las herramientas para desarrollar aplicaciones web o híbridas como son PhoneGap, Fabric, Ionic, App Inventor y los IDE's Android Studio de Google y XCode de Apple.

Aunque Android y iOS no son los únicos Sistemas Operativos Móviles, en este trabajo solo serán considerados estos dos, pues juntos abarcan más del 95% de la participación en el mercado mundial, los otros están dejando de ser usados, como BlackBerry y Symbian, otros son muy nuevos o aún no han tenido mucha participación en el mercado, como Firefox Mobile y Windows Phone.

Lo primero es aclarar cuáles son los diferentes tipos de aplicaciones para dispositivos móviles, entre las cuales tenemos aplicaciones web, aplicaciones nativas y aplicaciones híbridas. Las siguientes tres definiciones fueron tomadas y semi parafraseadas desde el sitio web:

<http://www.accensit.com/index.php/en/accensit-blog-en/150-mobile-platforms.html>

2. APLICACIONES WEB.

El desarrollo de Aplicaciones Web para dispositivos móviles es el desarrollo de páginas web que son optimizadas para ser visualizadas en las pantallas de dispositivos móviles y para ser utilizadas en pantallas táctiles. La aplicación se visualizará de forma casi idéntica en todos los dispositivos, dependiendo de la resolución de la pantalla. Las tecnologías utilizadas son las mismas que para un sitio web; se utiliza HTML5, CSS3 y JavaScript.

El usuario accede a estas aplicaciones mediante el navegador web de su dispositivo, aunque en función del dispositivo puede crear un enlace directo en su escritorio y acceder a ella como si se tratara de cualquier otra aplicación.

No se distribuyen mediante sitios oficiales de Apple Store o Google Play lo que permite zafarse de las tasas y limitaciones impuestas por estos. Funcionan principalmente online pero pueden contar con caché local y almacenamiento de datos local para el funcionamiento offline.

Ejemplos de Aplicaciones web móviles: Financial Times (web app con caché local), Yahoo Mail. Se acceden mediante un navegador web del dispositivo.

2.1 VENTAJAS DE LAS APLICACIONES WEB

- Costo: Se desarrolla una única aplicación para todos los sistemas operativos a diferencia de una aplicación nativa.
- Rapidez de desarrollo: Al desarrollar para todas las plataformas a la vez no se multiplica el tiempo de desarrollo.
- Compatibilidad: Se reutiliza casi en un 100% el mismo código fuente para todos los sistemas operativos de los dispositivos móviles. Se puede optimizar para las distintas resoluciones de pantallas móviles.
- Mantenimiento de un único código para todas las plataformas.
- Distribución: Se puede evitar la subida a los repositorios oficiales online (App Stores). Es posible descargar la aplicación desde una página web o accediendo a una URL concreta.
- Actualizaciones inmediatas: Las actualizaciones de la aplicación son inmediatas, a diferencia de unas semanas de espera de aprobación por Apple Store.
- Visibilidad: La URL de la aplicación web es fácil de compartir, e incluso se puede redirigir el tráfico desde redes sociales, blogs, códigos QR y medios de publicidad.

2.2 DESVENTAJAS DE LAS APLICACIONES WEB

- Integración con los componentes nativos del dispositivo: Las aplicaciones web, al ser ejecutadas en el navegador, solo pueden acceder a determinados componentes y funcionalidades nativas del dispositivo, aunque las posibilidades evolucionan con rapidez.
- Diferencias entre dispositivos. El nivel de incorporación de los estándares en los navegadores de cada dispositivo es bastante heterogéneo. Es posible que se dedique más tiempo en testear y corregir problemas de una aplicación web debido al comportamiento distinto de los navegadores que en el propio desarrollo de la misma.
- El uso offline: Las capacidades de almacenamiento local de las aplicaciones Web son limitadas, solicitándose al usuario la confirmación en caso de sobrepasar ciertos límites.
- Fluidez: Un interface de usuario desarrollado en HTML5 no es tan fluido como uno nativo.

3. APLICACIONES NATIVAS.

Para cada sistema operativo o plataforma móvil existe una herramienta de desarrollo distinta, esta herramienta incluye un IDE y un lenguaje de programación, es decir, si queremos desarrollar una misma app para

distintos Sistemas Operativos Móviles (en adelante SO) hay que desarrollar una aplicación distinta para cada uno, una para Android usando Android Studio y Java, una para iOS, usando XCode con Swing o Adjective-C. Entre las características más importantes de este tipo de aplicaciones están la posibilidad de acceder a otras aplicaciones nativas y controlar todos los recursos del dispositivo, por mencionar algunos: la cámara, los contactos, el GPS, el NFC, entre otros. También es importante mencionar que estas aplicaciones se instalan en el disco duro o tarjeta de memoria del dispositivo móvil.

3.1 VENTAJAS DE LAS APLICACIONES NATIVAS

- Posibilidad de utilizar funcionalidades del teléfono: cámara, GPS, contactos, acelerómetro, etc.
- Rendimiento: Es más rápida que una aplicación híbrida y mucho más rápida que una Web App, puede interactuar con otras aplicaciones nativas, saca mejor partido a los recursos del teléfono, uso de CPU, memoria y batería óptimo.
- Existe un sitio para compartir y comercializar las aplicaciones: Apple Store, Google Play según el caso.
- Funciona en modo online y offline: Se puede utilizar sin la necesidad de una conexión a internet.
- El proceso de verificación y aceptación por parte de Apple o Google suele ser más rápido que en una aplicación híbrida.
- Almacenamiento local seguro.

3.2 DESVENTAJAS DE LAS APLICACIONES NATIVAS

- El costo, aunque si se precisa el desarrollo de la aplicación para una única plataforma entonces suele ser más barato.
- Tiempo de desarrollo es mayor, en el caso que deba desarrollarse una aplicación para cada plataforma.
- Mayor Coste de mantenimiento: Mantener varias versiones de la misma aplicación para distintas plataformas es costoso.
- Tiempo de aprobación: La aprobación de una nueva aplicación en el Apple Store puede llegar a tardar un par de semanas. En Google Play son 24 horas.
- Cualquier modificación, actualización o nueva funcionalidad de la aplicación tiene que pasar por el mismo sistema de aprobación.
- Coste adicional: Solo si se desea publicar en la tienda oficial, Apple Store cobra 99 dólares cada año por cada aplicación publicada, Google Play cobra 25 dólares una única vez por cada aplicación que se publique, además, ambos cobran por cada aplicación disponible en su tienda 30%

del precio de la aplicación. Así como un 30% de los ingresos obtenidos a través de la aplicación mediante in-app purchase (compras desde la propia aplicación).

4. APLICACIONES HÍBRIDAS

Una aplicación híbrida o multiplataforma, como su nombre bien indica, es una “mezcla entre una aplicación Nativa y una WebApp”. Este tipo de aplicaciones se hicieron populares gracias al framework PhoneGap; pero existen varios que pueden ser utilizados: Kendo UI Mobile, Sencha Touch, Trigger.io o Titanium Appcelerator, Fabric, entre otros.

Al desarrollar aplicaciones híbridas utilizamos la tecnología nativa (conjunto de APIs) cuando es necesario o más nos conviene (para acceder a cámara, acelerómetro, contactos, etc.), y la tecnología web (como HTML5, CSS3 y JavaScript) para el desarrollo de la estructura e interfaz de la aplicación. De este modo maximizamos la base de código que es común a las distintas plataformas y limitamos el desarrollo de funcionalidad nativa a aquellos aspectos que no puedan ser desarrollados de otro modo.

4.1 VENTAJAS DE UNA APLICACIÓN HÍBRIDA

- Minimizamos el código específico: La mayor parte del código puede utilizarse para el resto de plataformas. Solo se utiliza código nativo para aquellos aspectos que lo requieran.
- Menor coste de desarrollo sobre todo si se requiere la aplicación en varias plataformas.
- Menor coste de mantenimiento al ser la mayor parte del código común a todas las plataformas.
- Una aplicación Híbrida puede acceder a los recursos del dispositivo móvil prácticamente como una nativa.
- Se distribuye mediante los respectivos “stores”.

4.2 DESVENTAJAS DE UNA APLICACIÓN HÍBRIDA

- Rendimiento: El rendimiento y la experiencia de usuario no pueden alcanzar los niveles de la aplicación nativa. Incluso los dispositivos más actuales tienen problemas gestionando interfaces desarrollados en HTML5.

- Habitualmente los procesos de aprobación en los correspondientes "stores" para aplicaciones híbridas son más estrictos y pueden llegar a ser rechazadas si no queda clara la funcionalidad proporcionada mediante la carga de código remoto.

5. PHONEGAP

PhoneGap (<http://phonegap.com/>) es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems. PhoneGap permite a los programadores desarrollar aplicaciones para diversos sistemas operativos de dispositivos móviles, utilizando herramientas de desarrollo web tales como JavaScript, HTML5 y CSS3.

Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo; ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema, pero no se tratan tampoco de aplicaciones web; teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo.

En la tercera versión de PhoneGap se incorpora el uso de una interfaz de comandos a través de consola, una nueva arquitectura de complementos descentralizados y la posibilidad de utilizar un código web unificado para crear múltiples proyectos.

PhoneGap maneja API's que permiten tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas en Javascript.

Existe una confusión con los nombres PhoneGap y Apache Córdoba, se puede decir que un mismo producto pero tiene dos nombres, esta confusión sucede porque la empresa que desarrollo este producto (Nitobi), decidió en el año 2011 donar a la Fundación Apache su código fuente, con la intención de fomentar el uso de la herramienta, pero posteriormente decidieron vender toda la empresa a Adobe, incluido su producto PhoneGap, y aunque Adobe permitió que este software continuara siendo libre, la fundación Apache decidió cambiar el nombre por Apache Córdoba, pues saben que PhoneGap es una marca registrada de Adobe y este puede en un futuro tener fines comerciales a este producto. Al día de hoy son herramientas iguales, que se pueden descargar desde dos sitios diferentes (<http://phonegap.com/> y <http://cordova.apache.org/>) y que sirven para lo mismo.

5.1 VENTAJAS

- Facilidad. Si conocemos lenguajes Web (especialmente JavaScript), desarrollar una aplicación con PhoneGap nos resultará mucho más fácil que hacerlo a través de los SDKs asociados a cada plataforma.
- Velocidad de desarrollo. Con PhoneGap, codificamos la aplicación una sola vez y la adaptamos a múltiples plataformas, en vez de escribir una aplicación diferente para cada una.
- Aplicación nativa. Nuestra aplicación tendrá todas las ventajas de las aplicaciones nativas.
- Libre y gratuito. PhoneGap es open source y se puede utilizar sin costo para una primera aplicación.
- Existen muchos plugins para extender las funcionalidades de PhoneGap y acceder a características adicionales del dispositivo.
- Debido a que las funcionalidades disponibles para PhoneGap son más limitadas respecto a las disponibles en plataforma nativa, se requieren menos habilidades de implementación sobre aplicaciones móviles dentro del equipo de desarrollo.

5.2 DESVENTAJAS

- Malas prácticas. Muchos desarrolladores web se acostumbran a crear aplicaciones nativas con PhoneGap, sin preocuparse por aprender a utilizar el lenguaje propio de cada plataforma. Así, no están en condiciones de desarrollar aplicaciones puramente nativas cuando realmente es la mejor opción.
- Mal rendimiento. Muchos desarrolladores de PhoneGap descubren que sus aplicaciones tardan demasiado en cargar. Esto puede deberse a la pesada estructura del framework y la necesidad de usar JavaScript para implementar efectos visuales complejos. Las imágenes también influyen considerablemente. Por eso, puede no ser una buena opción para desarrollar videojuegos.
- Limitaciones. PhoneGap no permite manipular las características del dispositivo con tanta libertad como el SDK correspondiente. Esto puede ser un problema si la aplicación a desarrollar tiene una alta complejidad.
- Inconsistencias. Cada sistema operativo tiene su propio diseño. Y una aplicación nativa debería integrarse visualmente con el sistema operativo donde se ejecuta. Pero esto es muy difícil cuando la interfaz es la misma para todas las plataformas.

5.3 LICENCIA

PhoneGap es Libre y gratuito, es open source y se puede utilizar sin costo para una aplicación, además de esto, si algún día Adobe decide cambiar el tipo de licencia, se seguirá contando con Apache Cordova, que es totalmente gratuita y por la ideología de la Fundación a que pertenece seguirá siendo libre, y que como se explicó anteriormente, PhoneGap y Apache Córdoba son el mismo software.

6. FABRIC

La gente Twitter, después de crear el famoso framework CSS llamado Bootstrap, ha desarrollado este nuevo framework con el que pretenden revolucionar como se desarrollan las aplicaciones móviles. Fabric (<https://get.fabric.io/>) es una plataforma móvil modular que permite a los desarrolladores construir aplicaciones móviles multi-plataforma y que, según sus creadores, ayuda a simplificar procesos en el desarrollo y a “crear las mejores aplicaciones con el mínimo esfuerzo”.

La plataforma Fabric está formada por tres kits modulares, y los SDKs que abordan los desafíos más comunes de los desarrolladores: estabilidad, distribución, ingresos y autenticación.

Estos tres kits combinan los servicios de Twitter, MoPub y Crashlytics. La idea es poder crear aplicaciones más estables, generar ingresos a través del intercambio de publicidad y sacar partido de los sistemas de autenticación y streaming en tiempo real de Twitter para una mayor distribución y una autenticación más simple.

Tiene plugins para los principales IDEs tanto iOS como Android (Xcode, Eclipse, IntelliJ, Android Studio) y está formado por una serie de módulos que el usuario podrá integrar a su conveniencia para conseguir la app que desee. La instalación se resume en una cuestión de gestionar los SDKs. Puedes instalar el kit que quieras según tus necesidades.

6.1 VENTAJAS

Al tratarse de aplicaciones híbridas, hereda todos los beneficios que caracterizan a estas, pero además:

- Permite hacer autenticación con Twitter mucho más simple e integra otros servicios de Twitter.
- Cubre los retos más importantes de los desarrolladores, estabilidad, distribución, ingresos e identidad.
- Combina los servicios de Crashlytics, MoPub, Twitter y otros para ayudar a desarrollar aplicaciones más estables, generar ingresos a través de la

mayor bolsa de anuncios móviles del mundo, lograr una mayor distribución al aprovechar los sistemas de inicio de sesión de Twitter y los flujos de contenido en tiempo real, y obtener una identidad más sencilla

- La instalación sólo lleva unos minutos
- La mayoría de las funciones sólo requieren unas pocas líneas de código para dedicar menos tiempo a gestionar los SDKs (por sus siglas en inglés, software development kit) y más tiempo desarrollando mejores experiencias para tus usuarios.

6.2 DESVENTAJAS

- Poco rendimiento: las aplicaciones que usan herramientas web y APIs nativas no son tan rápidas como lo son las puramente nativas.
- Limitaciones. Fabric no permite aun manipular todas las características de los dispositivos.

6.3 LICENCIA

Fabric es gratis, cualquier persona puede descargarlo y usarlo sin restricción alguna, además de los plugins para los IDE's más populares, se puede descargar una versión para dispositivos móviles desde la AppStore de Apple y Google Play de Google.

7. IONIC

El tercer framework que vamos a ver es Ionic (<http://ionic.io/>), que igual que los dos mencionados anteriormente es un Framework para desarrollar aplicaciones híbridas.

Ionic es una herramienta gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JavaScript. Está construido con Sass y optimizado con AngularJS, con la finalidad de construir aplicación muy interactivas. Algo importante es que se instala sobre Apache Córdoba o PhoneGap.

7.1 VENTAJAS

- Alto rendimiento: La velocidad es importante, Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.
- AngularJS & Ionic: Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.
- Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Pero con la ventaja de que desarrollas una vez, y compilas para varios SO.
- Bonito diseño: Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.
- Plugins: Este framework usa sus propios plugins y además puede usar los de Apache Córdoba.
- Dotado de una herramienta llamada Ionic Creator, que te permite crear aplicaciones basadas en este framework pero usando un entorno totalmente gráfico.

7.2 DESVENTAJAS

- No tiene acceso directo a las funciones de cámara, geolocalización, y otros, es decir, a los componentes hardware del dispositivo, lo cual se soluciona mediante plugins.
- Puede ser lento para mapas como Google Maps.
- No presenta temas para las diferentes plataformas, es decir, el look & feel no simula al nativo.
- Está construido bajo AngularJs, por lo que es necesario conocer previamente AngularJs.

7.3 LICENCIA

Ionic es de código abierto publicado bajo una licencia MIT. Esto significa que se puede utilizar en nuestros proyectos personales o comerciales, de forma gratuita. MIT es la misma licencia que utiliza otros proyectos populares como jQuery y Ruby on Rails.

8. APP INVENTOR.

App inventor es un framework creado inicialmente por el MIT (Instituto tecnológico de Massachusetts) y fué cogido por google, para que cualquier

persona con interés pueda crear su propia aplicación móvil, ya sea para su empresa, para su casa o por otros intereses.

Google App Inventor es una plataforma de Google Labs para crear aplicaciones de software para el sistema operativo Android. De forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones fruto de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

8.1 VENTAJAS

- Permite crear aplicaciones por medio de bloques de manera intuitiva y gráfica, sin necesidad de saber código de programación.
- Se puede acceder en cualquier momento y cualquier lugar siempre que estemos conectados a internet a nuestros proyectos.
- Nos ofrece varias formas de conectividad: directa, o WiFi o por medio del emulador.
- permite descargar el instalador de la aplicación mediante el archivo apk para compartirla o instalarla.

8.2 DESVENTAJAS

- No genera código el Java para posteriores desarrollos más profundos.
- No maneja mucha flexibilidad como lo hace el código en Java.
- Solo funciona con conexión a Internet.
- Solo se puede desarrollar para Android.

8.3 LICENCIA

Google App Inventor cuenta con una licencia libre, cualquiera puede simplemente registrarse en su web y empezar a desarrollar aplicaciones móviles que después ejecutará en Android sin ningún problema.

9. ANDROID STUDIO

Esta es la herramienta oficial para la creación de aplicaciones nativas para el SO Android, cuenta con todas las herramientas necesarias para desarrollar todo tipo de aplicaciones para este SO, en la actualidad se encuentra en su versión 1.5.1, la cual puede ser descargada desde:

<http://developer.android.com/intl/es/sdk/index.html>.

Android Studio es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionaran las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de plugins).

Cuenta con muchas características nuevas e innovadoras, entre las que podemos mencionar:

- Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna).
- Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.
- Nueva interfaz específica para el desarrollo en Android.
- Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.
- Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.

- Integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo xml.

9.1 VENTAJAS

Cuenta con todas las ventajas de las aplicaciones nativas, pero además cuenta con muchas otras características, entre las que podemos mencionar las siguientes:

- Android Studio es el futuro del desarrollo de aplicaciones para android.
- Multiplataforma, se puede instalar en Linux, Mac OS y en Windows.
- Interfaz intuitiva.
- Integración de dispositivo virtual, posibilidad de emular diferentes dispositivos.
- Creación de elementos gráficos sin uso de código.
- Aplicaciones nativas con control total del hardware del dispositivo.
- Integración de herramientas de control de versiones como Git, SVN, y el repositorio GitHub.
- Es el IDE oficial de la empresa que desarrolla Android.

9.2 DESVENTAJAS

- La primera ejecución del Emulador es algo lenta.
- No se pueden desarrollar aplicaciones para otros SO móviles.

9.3 LICENCIA

Cuenta con una licencia Apache 2.0, lo que significa que es libre, podemos acceder y el software sin ninguna restricción y sin pagar por su uso.

10. XCODE

Xcode es el entorno de desarrollo integrado que nos permite crear aplicaciones nativas para los dispositivos de la marca Apple, este IDE se suministra gratuitamente junto con Mac OS X. Xcode trabaja conjuntamente con Interface

Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario.

Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C, C++, Swift, Objective-C, Objective-C++, Java y AppleScript mediante una amplia gama de modelos de programación, incluyendo, pero no limitado a Cocoa, Carbón y Java. Otras compañías han añadido soporte para GNU Pascal, Free Pascal, Ada y Perl.

Entre las características más apreciadas de Xcode está la tecnología para distribuir el proceso de construcción a partir de código fuente entre varios ordenadores, utilizando Bonjour.

10.1 VENTAJAS

- Es el IDE oficial de la empresa que desarrolla iOS.
- Permite desarrollar todo tipo de app para dispositivos iOS.
- Permite tener control total de los recursos hardware del dispositivo.

10.2 DESVENTAJAS

- Solo se puede ejecutar en una plataforma, Mac OS.
- Solo se pueden realizar app para los dispositivos de la misma marca, es decir, es un software totalmente restrictivo.
- NO usa emulación sino simulación, esto causa fallas al dar una representación exacta de un dispositivo iOS.

10.3 LICENCIA

El software se entrega “gratuitamente” con la compra de un computador de la misma marca, pero además de esto se debe adquirir una licencia de desarrollador para poder probar y subir a la tienda las app desarrolladas. Así que finalmente no es tan gratuito

11. CARACTERÍSTICAS A EVALUAR

Para determinar cuál de las herramientas antes mencionadas usar en este proyecto, hemos decidido evaluar algunas características de cada una de ellas, y asignarles una calificación de 1 (uno) a 5 (cinco) puntos y la que acumule mayor cantidad de puntos será la ganadora, las características a evaluar son: rendimiento, curva de aprendizaje, conocimientos actuales y costo de licencias.

11.1 RENDIMIENTO

Se refiere a la capacidad de ejecución de forma rápida de la aplicación desarrollada con esta herramienta, donde 1 (uno) es muy lenta y 5 (cinco) es muy rápida.

11.2 CURVA DE APRENDIZAJE

Esta característica se refiere a que tan fácil es aprender a usar la herramienta, para esto 1 (uno) es muy difícil de aprender y 5 (cinco) es muy fácil de aprender

11.3 CONOCIMIENTOS ACTUALES

Hace referencia a los conocimientos que el equipo de desarrollo tiene sobre los lenguajes o tecnologías necesarias para trabajar con esta herramienta, donde 1 (uno) es no sabemos nada del tema y 5 (cinco) es somos expertos en dichas tecnologías.

11.4 COSTO DE LICENCIAS

Esta característica hace mención al costo de comprar licencias para el desarrollo y los costos de publicar y mantener la app en las tiendas oficiales, donde 1 (uno) es no cuesta nada y 5 (cinco) es cuesta mucho.

12. COMPARACIÓN ENTRE FRAMEWORKS

En la siguiente tabla se muestra la valoración que le asignamos a cada característica,

| Característica | PhoneGap | Fabric | Ionic | App Inventor | Android Studio | Xcode |
|------------------------|-----------|-----------|-----------|--------------|----------------|-----------|
| Rendimiento | 3 | 3 | 4 | 3 | 5 | 5 |
| Curva de Aprendizaje | 3 | 3 | 3 | 3 | 2 | 2 |
| Conocimientos Actuales | 2 | 2 | 2 | 2 | 4 | 2 |
| Costos de Licencias | 4 | 4 | 4 | 4 | 4 | 1 |
| Total | 12 | 12 | 13 | 12 | 15 | 10 |

13. SELECCIÓN DE FRAMEWORK

Según las calificaciones asignadas, la herramienta seleccionada es el Android Studio, que aunque tiene una curva de aprendizaje larga, lo favorece el rendimiento, el costo de licencias y que nosotros ya manejamos muchas de las tecnologías, como es el lenguaje de programación Java y los archivos XML usados en las vistas.

14. PRUEBA DE CONCEPTOS

Al iniciar la aplicación, se muestran cuatro de las seis secciones en las que está dividida la app, estas opciones se podrán explorar sin necesidad de autenticación, lo cual favorece una experiencia de usuario positiva, en parte inferior se encuentra un botón de doble estado (ToggleButton) con el cual se puede cambiar a la vista del administrador.



Para probar la vista que desarrollamos para este trabajo, es necesario pasar a esta vista y seleccionar la sección canchas, donde se podrá ver el formulario para agregar una nueva cancha.

Cabe anotar que las imágenes usadas son imágenes de referencia y no necesariamente serán usadas en futuras versiones de la aplicación.

15. BIBLIOGRAFÍA

<http://www.campusmvp.es/recursos/post/PhoneGap-o-Apache-Cordova-que-diferencia-hay.aspx>

<http://www.genbetadev.com/desarrollo-aplicaciones-moviles/fabric-la-suite-para-desarrollo-de-apps-moviles-de-twitter>

<https://blog.twitter.com/es/2014/introduciendo-fabric>

<http://revolucion.mobi/2014/11/01/tutorial-mi-primera-app-con-ionic/>

<http://jesusmur.com/porque-desarrollar-apps-moviles-con-ionic-framework/>