UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

FACULTAD DE CIENCIAS Y TECNOLOGÍA

DEPARTAMENTO DE COMPUTACIÓN

UNAN- LEÓN

Monografía para optar al título Ingeniero en Sistemas de Información.

DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA MEDIANTE EJEMPLO PROPUESTO CON EL ENTORNO DE DESARROLLO CORONA SDK.

Autores:

- Antonio José Méndez González.
- José Dolores Hernández Betanco.
- Eliezer Esteban Gutiérrez Ruiz.

Tutor: Ing. Otón Jossemar Castillo Navas.

León, Mayo 2014.

“A La Libertad Por La Universidad”

AGRADECIMIENTO

Expresamos nuestro profundo agradecimiento, a:

- El Creador de todas las cosas, nuestro Dios todo poderoso, por habernos dado la fuerza y sabiduría para llegar a culminar con éxito, el presente trabajo monográfico.

- La Universidad Nacional Autónoma de Nicaragua, UNAN- León, por habernos acogido durante estos cinco años de arduo esfuerzo, garantizándonos nuestra formación profesional.

- Nuestro tutor, Ing. Otón Jossemar Castillo Navas, por habernos guiado con sus conocimientos y experiencia, hasta finalizar este trabajo monográfico.

Los Autores.

DEDICATORIA

Dedicamos el presente trabajo monográfico, fundamentalmente a:

- Dios Padre, por habernos dado la vida y fuerza para culminar nuestra investigación monográfica.

- Nuestras Madres y Padres: Graciela Méndez, Lucia Betanco, Ernesto Hernandez, Martha Ruiz, Esteban Gutiérrez y Darling López por habernos brindado apoyo moral, espiritual y económico, durante cinco largos años de estudio.

- A nuestros maestros que en este andar por la vida, influyeron con sus lecciones y experiencias en formarnos como una persona de bien y preparada para los retos que pone la vida.

Los Autores.

INDICE

1. INTRODUCCIÓN

Actualmente la tecnología juega un papel fundamental en nuestra sociedad, hasta el punto de que no se podría entender la vida actual sin el uso de ordenadores. La aparición de las nuevas tecnologías en la sociedad supuso la agilización, optimización y el perfeccionamiento de las actividades cotidianas.

Sin duda uno de los campos que más ha notado estas mejoras es el de la comunicación, a través de Internet podemos comunicarnos de forma rápida en cualquier parte del mundo y durante las 24 horas del día.

Es tan importante la presencia de dispositivos electrónicos en nuestra vida, que aunque quizás no nos detenemos a pensar en ello, siempre llevamos uno. Y es que la actual demanda tecnológica es sin duda el mundo de los dispositivos móvil e Internet.

Durante mucho tiempo estos dos términos han estado separados, el teléfono móvil se trataba de un aparato que sólo servía para contactar mediante llamadas y mensajes con otras personas. Ahora estos dos mundos van de la mano y el teléfono móvil se ha convertido en un dispositivo indispensable por muchos usuarios para acceder a Internet y realizar numerosas actividades.

Este elevado uso de los dispositivos móviles para acceder a Internet es sin duda porque ofrecen una serie de prestaciones que no conseguimos con los ordenadores.

Evidentemente una de las grandes ventajas es la movilidad. Su tamaño junto con la autonomía que proporcionan las baterías permite al usuario poder transportarlos.

También ofrecen varias formas para acceder a Internet: la gran mayoría cuentan con conexión wifi, pero también con el sistema 3G o 4G. Muchos de los usuarios

reconocen que incluso estando en su domicilio utilizan los dispositivos móviles, quizás tenga mucho que ver la posibilidad de conectarse a la red wifi de su hogar.

Desde hace tiempo para cualquier empresa es imprescindible tener presencia en la red, sin embargo hoy en día ya no es suficiente, ya que operadores, fabricantes y desarrolladores compiten en la conquista de un mercado que no deja de crecer; para lograrlo muchas empresas han tenido que desarrollar sus propios sistemas operativos para los Smartphone tales como lo son Android, iOS, Windows Phone, Symbian, entre otros, que a su vez cada uno de estos ha ido superando uno al otro poniéndose a la vanguardia Android e iOS como mayores competidores del mercado, teniendo como punto de referencia el buen diseño, funcionalidad, facilidad de uso, una variedad de aplicaciones, juegos enorme y su perfecta integración con servicios en la nube lo que ha hecho que estas empresas obtengan un ingreso enorme de capital tanto como un prestigio.

Como estas son empresas independientes compitiendo por ser la mejor ha causado que el mercado este dividido para los usuarios lo cual dificulta la elección de un Smartphone ya que estos no cuentan con toda la gama de aplicaciones que se desarrollan en cada una de ellas. Lo que conlleva a pérdidas monetarias en estas grandes empresas y motiva a implementar un nuevo método del cual los usuarios puedan beneficiarse de manera automática de todas estas compañías.

Por ello hemos decidido implementar el desarrollo de aplicaciones móviles multiplataforma mediante el entorno Corona SDK y el lenguaje de programación Lua. Creando una aplicación, la cual será un libro de diversas recetas de cocinas que será de utilidad para cualquier persona, debido a que la tecnología avanza, las necesidades de las personas se han vuelto más exigentes por lo que requieren de respuestas inmediatas, así demostrando cuales son las facilidades de implementar un entorno multiplataforma para dar respuesta a diferentes necesidades con un solo método de programación.

La programación para plataformas diferentes (Android e iOS) hace que los costes de producción de aplicaciones móviles se disparen y que en algunos casos el desarrollo de las aplicaciones no sea rentable. En estos casos siempre se tiende a desarrollar una única versión de la App y perder una cuota de potenciales usuarios importante con tal de reducir gastos.

2. ANTECEDENTES

La forma en que la sociedad se conecta a la red, ha cambiado radicalmente en los últimos años. A ello ha contribuido en parte, la gran cuota de mercado que están adquiriendo los llamados teléfonos inteligentes o Smartphone, vemos cómo las conexiones a internet a través de estos dispositivos han mejorado su calidad, mientras que sus precios se han disminuido drásticamente.

La proliferación de dispositivos móviles y plataformas representan un cambio tecnológico que afecta a diferentes niveles. Las empresas deben decidir, no sólo el mejor uso estratégico de las plataformas móviles, también, de qué manera su aplicación será más eficiente. Inevitablemente, esta tarea recae en los desarrolladores, que se enfrentan al desafío de desarrollar las aplicaciones móviles. Un desafío nada fácil, teniendo en cuenta el número de dispositivos móviles y plataformas que existen.

Debido a estas constantes actualizaciones tecnológicas que han surgido cabe la necesidad desarrollar aplicaciones que faciliten la vida diaria de las personas, considerando que existen variedad de dispositivos móviles con sistemas operativos diferentes, lo cual dificulta la versatilidad de las aplicaciones, para ello tendríamos que desarrollar una aplicación con lenguaje de programación diferente por cada sistema que exista, lo que causa que haya pérdida de tiempo y sobre todo de dinero para cualquier compañía además de que tendríamos que dominar todos los lenguajes o por lo contrario contratar a un programador por cada tipo de lenguaje, por ende no es viable ya que incurre en demasiados gastos.

A la hora de desarrollar aplicaciones móviles las empresas están teniendo que enfrentarse a los problemas de la segmentación de los dispositivos y les está obligando a desarrollar la misma aplicación varias veces con los lenguajes nativos de cada sistema es decir, en diferentes plataformas para cubrir una amplia cuota de mercado, gastando tiempo y recursos por cada dispositivo en el que se quiera incursionar.

3. JUSTIFICACIÓN

Con el incremento exponencial de dispositivos y sistemas operativos se hace necesario el desarrollo de App que funcionen en diversos SO para abarcar una mayor cuota de usuario y tener un mejor mercado lo que acarrea la necesidad de buscar herramientas que faciliten el desarrollo de estas.

Este tipo de tecnología a implementar al momento de crear una aplicación móvil son muy oportunas y necesarias para la efectividad de la misma, ya que el desarrollar en multiplataforma ahorra tiempo que se usa para el desarrollo y estando basada en diferentes lenguajes de programación la fecha de publicación sería una variante más a tomar en cuenta, debido a que por cada plataforma la reutilización del código es mínima y la extensión de líneas de código es variables en diferentes lenguajes. Corona SDK es una opción de las más completa, abarcando con su lenguaje de programación LUA las 2 principales plataformas móviles con más de un 95% de dispositivos, generando aplicaciones nativas aumentando la eficiencia y aprovechando mejor los recursos con los que se dispone así como las Apis nativas haciendo que la experiencia del usuario sea más agradable y mucho más fluidas que aplicaciones hibridas y con el entorno de desarrollo Corona SDK generamos los archivos ejecutables .apk e .ipa para Android e iOS respectivamente.

Otras de las ventajas es el hecho de disminuir el costo monetario para las empresas ya que solo se necesita el conocimiento sobre un solo lenguaje para la creación de otras aplicaciones con sistemas diferentes sin necesidad de contratar a varios programadores que dominen un lenguaje especifico haciendo que haya ahorro de tiempo, espacio y dinero.

4. OBJETIVOS

4.1 Objetivo General

- Demostrar la funcionalidad y factibilidad de desarrollar aplicaciones multiplataforma mediante el entorno de desarrollo Corona SDK

4.2 Objetivos específicos

- Conocer las ventajas de Corona SDK.

- Aplicar los paradigmas y características del lenguaje Lua necesario en Corona SDK.

- Implementar el desarrollo de una aplicación en multiplataforma con Corona SDK.

5. METODOLOGÍA

5.1 Metodología de trabajo

La metodología empleada para el trabajo monográfico es Iterativa Incremental debido a que nuestro proyecto lo desarrollamos en etapas temporales para llegar a un resultado final tomando en cuenta cada uno de estas etapas anteriores como punto de partida.

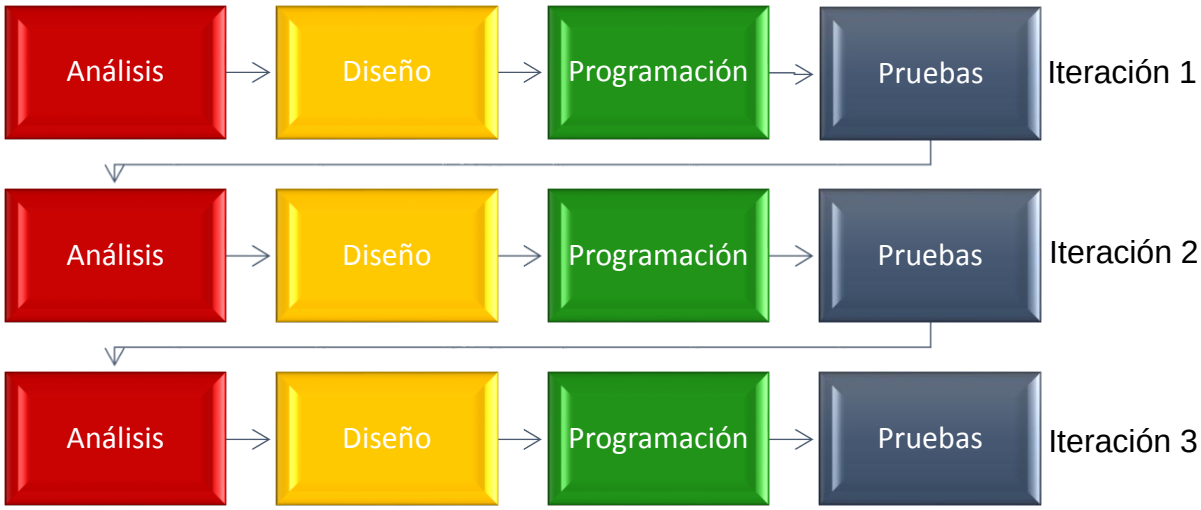
El Modelo Incremental es de naturaleza interactiva brindando al final de cada incremento un resultado de proyecto a desarrollar completamente operacional. Este modelo nos es particularmente útil porque no contamos con una cantidad grande de personas trabajando sobre el proyecto, lo que permite que desarrollemos éste en etapas funcionales para evitar riesgos técnicos y así poder obtener un resultado óptimo al finalizarlo

El propósito de este método es la de sacar la mayor ventaja posible de cada iteración que realicemos en la implementación de sistema. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

Este proceso lo hemos realizado en cuatro etapas las cuales son:

- Análisis: analizamos las necesidades del usuario para determinar que funcionalidad debe cubrir. En esta etapa reuniremos todo lo que requiere el sistema de una iteración del desarrollo del sistema debido a que iremos descubriendo en cada proceso nuevas necesidades para las demás iteraciones.
- Diseño: diseñaremos la App que utilizara el usuario haciendo que esta sea atractiva, de interés y fácil manipulación para quien la utilice.
- Programación: codificaremos el diseño antes realizado para que pueda ser funcional.

- Pruebas: realizamos pruebas de la App para ratificar el correcto funcionamiento de esta.



5.2 Materiales Utilizados

Hardware: para la realización de la App a nivel hardware requerimos de ciertas herramientas.

- PC: 1.7 Ghz, Intel Dual Core
- Ram: 2 GB
- Disco Duro: 160 gb
- Dispositivos móviles. (Tablet, Smartphone)

Software: las herramientas software que requerimos son:

- Windows 7 (Sistema Operativo)
- Mysql
- SQlite
- Corona SDK
- Sublime text 3

6. MARCO TEÓRICO

6.1 Plataformas

Una plataforma es una combinación de hardware y software utilizado para ejecutar aplicaciones de software. Una plataforma puede ser descrita simplemente como un sistema operativo o arquitectura de ordenador, o podría ser la combinación de ambos.

Las plataformas de software pueden ser un sistema operativo o entorno de programación, aunque más comúnmente se trata de una combinación de ambos. Una notable excepción a esto es Java, que utiliza un sistema operativo independiente de la máquina virtual para cada código compilado, conocido en el mundo de Java como bytecode. Ejemplos de plataformas de software incluyen:

- Android (sistema operativo) para teléfonos inteligentes y Tablet PC
- iOS Sistema de Apple
- RIM Sistema de Blackberry
- Windows Phone sistema de Microsoft

6.2 Aplicaciones Nativas

Una App nativa es una aplicación implementada en el lenguaje nativo de cada terminal. Estas App podrán acceder a los sensores internos del móvil para aprovecharse de funcionalidades típicas de estos dispositivos como el geoposicionamiento, brújula, cámara, etc.

Las aplicaciones nativas están implementadas en el lenguaje nativo del propio terminal: Objective-C para iOS, java para Android o C# para Windows Phone son algunos de los lenguajes más importantes.

6.2.1 Ventajas

- La principal ventaja de una App nativa el poder beneficiarse de los canales de distribución de los Market Places de cada plataforma

- Cada plataforma cuenta con un servicio de recopilación y distribución de App tanto de pago como gratuitas.

- Todos los recursos del dispositivo estarán accesibles para poder sacar el máximo partido de la App.

- Al finalizar la instalación se dispondrá de un acceso directo para poder lanzar la aplicación de una forma fácil y rápida.

- Al estar instalada en el propio Smartphone no es necesario contar con una conexión a internet, si bien es cierto que algunas partes de la aplicación pueden requerir de dicha conexión.

6.2.2 Desventajas

- Para publicar la aplicación el desarrollador se enfrentará a los procesos de validación de los diferentes Market Places.

- Al tener que desarrollar específicamente para cada plataforma el tiempo de desarrollo y el coste se incrementarán.

- El usuario deberá actualizar manualmente la aplicación desde los Market Places.

- A la hora de publicar la aplicación el desarrollador se enfrentará a los procesos de validación de los diferentes Market Places, algunos más duros que otros.

- Al tener que desarrollar específicamente para cada plataforma el tiempo de desarrollo y el coste se incrementarán.

6.3 Multiplataforma

En informática, multiplataforma, es un atributo conferido a los programas informáticos o los métodos de cálculo y los conceptos que se ejecutan e interoperan en múltiples plataformas informáticas. Software multiplataforma puede dividirse en dos tipos; requiere una compilación individual para cada plataforma que le da soporte, y el otro se puede ejecutar directamente en cualquier plataforma sin preparación especial, por ejemplo, el software escrito en un lenguaje interpretado o pre-compilado portátil bycode para que los intérpretes o paquetes en tiempo de ejecución son componentes comunes o estándar de todas las plataformas.

Para que el software pueda ser considerado multiplataforma, debe ser capaz de funcionar en más de una arquitectura de ordenador o sistema operativo. Esto puede ser una tarea que consume tiempo, ya que los diferentes sistemas operativos tienen diferentes interfaces de programación de aplicaciones o API (por ejemplo, Linux utiliza una API diferente de Windows).

El hecho de que un determinado sistema operativo se pueda ejecutar en diferentes arquitecturas de computadora no quiere decir que el software escrito para ese sistema operativo automáticamente funcione en todas las arquitecturas que soporta el sistema operativo.

6.3.1 Ventajas

- Fácil diseño.
- Fácil implementación.
- Seguridad.

6.3.2 Desventajas

- No hay acceso completo a todas las APIs nativas del móvil.
- Lentitud cada petición que hagamos en nuestra aplicación implicará una recarga de la página o un acceso.
- Peor monetización

6.4 LUA

Lua es un lenguaje de programación extensible diseñado para una programación procedimental general con utilidades para la descripción de datos. También ofrece un buen soporte para la programación orientada a objetos, programación funcional y programación orientada a datos. Se pretende que Lua sea usado como un lenguaje de script potente y ligero para cualquier programa que lo necesite. Lua está implementado como una biblioteca escrita en C limpio (esto es, en el subconjunto común de ANSI C y C++).

Siendo un lenguaje de extensión, Lua no tiene noción de programa principal (main): sólo funciona embebido en un cliente anfitrión, denominado programa contenedor o simplemente anfitrión (host). Éste puede invocar funciones para ejecutar un trozo de código Lua, puede escribir y leer variables de Lua y puede registrar funciones C para que sean llamadas por el código Lua. A través del uso de funciones C, Lua

puede ser aumentado para abarcar un amplio rango de diferentes dominios, creando entonces lenguajes de programación personalizados que comparten el mismo marco sintáctico. La distribución de Lua incluye un programa anfitrión de muestra denominado Lua, que usa la biblioteca de Lua para ofrecer un intérprete de Lua completo e independiente.

Lua es software libre, y se proporciona, como es usual, sin garantías, como se establece en su licencia.

Este un lenguaje de programación imperativo, estructurado y bastante ligero que fue diseñado como un lenguaje interpretado con una semántica extendible. El nombre significa "luna" en portugués y gallego.

6.4.1 Historia

Lua fue creado en 1993 por Roberto Lerusalimschy, Luiz Henrique de Figueiredo y Waldemar Celes basado en C y Perl con una estructura similar. Miembros del Grupo de Tecnología en Computación Gráfica (Tecgraf) en la Pontificia Universidad Católica de Río de Janeiro. Las versiones de Lua anteriores a la 5.0 fueron distribuidas bajo una licencia similar a la BSD, de la versión 5.0 en adelante se utiliza la licencia MIT, compatible con la GPL. Lua ha sido usado en muchas aplicaciones comerciales y no comerciales, cuyo número incrementa cada año.

6.4.2 Características

Lua es un lenguaje de extensión, suficientemente compacto para usarse en diferentes plataformas. En Lua las variables no tienen tipo, sólo los datos y pueden ser lógicos, enteros, números de coma flotante o cadenas. Estructuras de datos como vectores, conjuntos, tablas hash, listas y registros pueden ser representadas utilizando la única estructura de datos de Lua: la tabla.

La semántica de Lua puede ser extendida y modificada redefiniendo funciones de las estructuras de datos utilizando metatablas, casi como en Perl. Lua ofrece

soporte para funciones de orden superior, recolector de basura. Combinando todo lo anterior, es posible utilizar Lua en programación orientada a objetos

6.4.3 Funcionamiento interno

Los programas en Lua no son interpretados directamente, sino compilados a código bytecode, que es ejecutado en la máquina virtual de Lua. El proceso de compilación es normalmente transparente al usuario y se realiza en tiempo de ejecución, pero puede hacerse con anticipación para aumentar el rendimiento y reducir el uso de la memoria al prescindir del compilador.

También es posible la compilación en tiempo de ejecución utilizando LuaJIT.

6.4.4 Potencias del Lenguaje

En principio, lo más destacable, es su tamaño, lo cual permite que sea el candidato perfecto para ser embebido dentro de una aplicación como puede ser un software de retocado de imágenes, ofimática o videojuegos (como Word of Worldcraft, Half-Life 2, etc.)

La sintaxis del lenguaje les será bastante reconocible a los que programen en Ruby, ya que es muy parecida, pero agrega otras cosas, como los bucles por rangos de Python (o shell script).

Otra cosa a denotar, también muy parecida a Ruby, son los closures, funciones que se pueden usar como variables; y la definición de objetos (y sus métodos) en tiempo de ejecución, llegando a poder definir una función dentro de una estructura de datos típica, como por ejemplo un hash.

6.4.5 Desventajas de los lenguajes interpretados

La ejecución del programa por medio de un intérprete es usualmente mucho menos eficiente que la ejecución de un programa compilado. No es eficiente en tiempo

porque, o cada instrucción debe pasar por una interpretación en tiempo de ejecución, o como en más recientes implementaciones, el código tiene que ser compilado a una representación intermedia antes de cada ejecución. La máquina virtual es una solución parcial al problema de la eficiencia del tiempo pues la definición del lenguaje intermedio es mucha más cercana al lenguaje de máquina y por lo tanto más fácil de ser traducida en tiempo de ejecución. Otra desventaja es la necesidad de un intérprete en la máquina local para poder hacer la ejecución posible.

6.4.6 Ventajas de los lenguajes interpretados

Los lenguajes interpretados dan a los programas cierta flexibilidad adicional sobre los lenguajes compilados. Algunas características que son más fáciles de implementar en intérpretes que en compiladores incluyen, pero no se limitan, a:

- Independencia de plataforma (por ejemplo el byte code de Java)
- Reflexión y uso reflexivo del evaluador (ej. una función eval de primer orden)
- Generación funcional de primer orden, y orden sin necesidad de especificar metadata
- Posibilidad de generación de código in-situ, sin necesidad de recurrir a una compilación (ie. Spring)
- Tipos Dinámicos
- Facilidad en la depuración (es más fácil obtener información del código fuente en lenguajes interpretados)
- Pequeño tamaño del programa (puesto que los lenguajes interpretados tienen flexibilidad para elegir el código de instrucción)

- Ámbito dinámico

6.4.7 Aplicaciones

Lua ha sido usado para procesar datos de entrada a sistemas complejos, configurar aplicaciones, controlar hardware y muchos otras cosas.

En el gestor de ventanas Ion es posible utilizar Lua para personalizar la apariencia y extender su funcionalidad.

El gestor de ventanas awesome en su versión 3 utiliza Lua para su fichero de configuración.

Lenguajes como ActionScript, Java, Lua y otros, son empleados en distintos sistemas operativos, lo cual consigue un ahorro de costes, al simplificar el trabajo de desarrollo de un nuevo programa de software, al añadirlos como partes "prefabricadas" que incluso al adaptar o portar el programa a nuevos usos, por ejemplo de plataformas de videoconsolas a sistemas operativos como Android y otros, no necesitan ser modificados o mínimamente, convirtiéndolo en un programa de software de calidad nuevo, a un coste de desarrollo muy reducido.

También Lua es uno de los lenguajes de programación más utilizados para homebrews de la consola PSP de Sony debido a su sencillez. Van desde aplicaciones para añadir complementos u otros programas fácilmente a la consola a entornos de ventanas excelentes y videojuegos muy completos.

Por otro lado es utilizado para los productos de la compañía canadiense desarrolladora de software Indigo Rose, en AutoPlay Media Studio; estos programas no fueron desarrollados en Lua, sino que utilizan Lua para generar y crear scripts, ya sea por un asistente o por la pericia del programador.

6.4.8 Portabilidad

Gracias a su sencillez, Lua pudo ser portado a múltiples arquitecturas fácilmente. Lo más sorprendente es que existen ports para PSP y Wii, siendo por ahora el primer port el más antiguo y exitoso, mientras que el de Wii deriva de éste.

6.5 CORONA SDK

6.5.1 Corona

Cuando se desarrolla una aplicación, en ocasiones, se agradece utilizar herramientas que nos permitan utilizar nuestro producto en diferentes entornos o plataformas heterogéneas. En ese caso, suele ser habitual sacrificar algunas funcionalidades que son exclusivas de un dispositivo concreto. Sin embargo, tenemos la gran ventaja de minimizar el coste de desarrollo pues podemos llevar nuestra aplicación a otros entornos o plataformas sin tener que reescribir el código

Corona SDK es un kit de desarrollo de software (SDK), creado por Walter Luh, Fundador de Corona Labs Inc. . Corona SDK permite a los programadores de software para construir aplicaciones móviles para iPhone, iPad y dispositivos Android.

Corona permite a los desarrolladores utilizar de manera integrada Lua , para crear aplicaciones gráficas. El SDK no cobra los derechos por cada aplicación o impone cualquier requisito de la marca, y tiene un modelo de compra basado en suscripción.

Corona SDK es precisamente eso, una herramienta para desarrollar aplicaciones y ejecutarlas en iOS, Android, Amazon Kindle Fire y Burnes&Noble Nook. El tipo de aplicaciones al que va destinado principalmente es a juegos, aunque también permite otras funcionalidades que podrían emplearse en otros ámbitos. El lenguaje que utiliza es Lua, un lenguaje de scripting utilizado precisamente para desarrollar juegos.

En el caso de videojuegos dispone funcionalidades interesantes como la animación de Sprites o la utilización de algoritmos de física que nos facilitan el desarrollo de los juegos tipo Angry Birds. También nos permitirá conectarnos a Facebook o Twitter para crear juegos sociales.

En otros ámbitos, también nos permite mezclar tecnología HTML5 con OpenGL o utilizar funcionalidades empresariales como acceder a una base de datos local SQLite, librerías JSON y conexión asíncrona HTTP para guardar datos en una nube.

También permite el acceso a dispositivos extras que suelen llevar la mayoría de los móviles como el GPS, la cámara, el álbum de fotos o el giroscopio. Finalmente, hay disponibles herramientas de terceros que añadidas al SDK de Corona permitirán ampliar funcionalidades concretas.

6.5.2 Historia

Walter Luh empezó Corona Labs después de partir de Adobe en 2007. En Adobe, Luh fue el arquitecto principal de trabajo en el equipo Flash Lite. En junio de 2009, Luh lanzó el primer beta de Corona SDK gratuito para los primeros adoptantes.

En diciembre de 2009, Corona Labs lanzó Corona SDK 1.0 para iPhone . El mes de febrero siguiente, el SDK Corona 1.1 fue lanzado con características adicionales.

En abril de 2010, la Corona SDK 2.0 beta fue lanzado. Nueva característica definitiva Corona 2.0 era soporte multiplataforma para el iPhone, iPad y dispositivos Android . Más tarde ese mes, Corona Labs anunció una versión beta de Corona Game Edition, que incluye un motor de física y otras características avanzadas dirigidas específicamente a desarrollo de juegos .

En enero de 2011, Corona SDK fue lanzado para Windows XP y posteriores, dando a los desarrolladores la oportunidad de crear aplicaciones Android en PC .

6.5.3 Las características principales

El SDK expone las características tales como:

- Audio y gráficos
- Criptografía
- Creación de redes
- Información del dispositivo, tales como información de acelerómetro, GPS, y la entrada del usuario.

6.5.4 ¿Por qué usar Corona?

- Desarrollar aplicaciones 10 veces más rápido. Corona posee una extensa biblioteca de API que permite desde la animación de redes, con unas pocas líneas de código. Si uno está construyendo juegos o aplicaciones de negocio , podrá ver los cambios al instante en Corona Simulator y puede repetir con extrema rapidez. El desarrollo se realiza en Lua, un lenguaje de programación veloz, fácil de aprender.

- Publicar en todas las principales plataformas. Corona SDK permite publicar para iOS, Android, Kindle Fire y el Nook de una sola base de código. Windows Phone 8 y Windows 8 son muy pronto. Ellos se encargaran del trabajo pesado en relación con el dispositivo y la fragmentación de la plataforma, lo que le permite centrarse en la creación de contenido móvil excepcional .

- Basarse en estándares de la industria. La plataforma Corona se basa en estándares de la industria incluyendo OpenGL, OpenAL, Box2D, Facebook, SQLite y mucho más. El nuevo motor de estado-of-the-art gráficos, basado

en OpenGL 2.0 y shaders, permite efectos cinematográficos con sólo unas pocas líneas de código.

- Monetizar, crecer y prosperar. Los deseos de Corona es que uno tenga éxito. Por eso apoyan las compras in-app, una variedad de redes de publicidad y otros modelos de monetización. Se asocian con líderes incluyendo Vungle, PlayHaven, SponsorPay, Amazon y Fortumo y muchos más, para ofrecerle una amplia gama de opciones.

- Próspero ecosistema de Corona. Soporte Comunidad - Los foros están llenos de desarrolladores ofreciendo consejos, compartiendo código y echar una mano. A nivel mundial - Unirse a un grupo Corona en su área para colaborar y trabajar en red con sus pares. Recursos - Poseen cientos de guías, tutoriales, vídeos y proyectos de ejemplo para que adquiera velocidad. 3ª Parte herramientas – La comunidad ha construido herramientas de ahorro de tiempo que complementan Corona SDK y proporcionan funcionalidad extra.

6.5.5 Corona SDK herramientas y servicios de terceros

Corona SDK incluye las siguientes herramientas y servicios de terceros:

- **Kwik** es un plugin que permite a Photoshop a los usuarios crear aplicaciones móviles para dispositivos iOS y Android con Corona SDK. Diseñadores gráficos, escritores e ilustradores pueden crear aplicaciones móviles en Photoshop, y publicar usando Corona SDK, sin necesidad de codificación.

- **Lua Glider** es un IDE que cuenta con un depurador, las variables de explorador, la pila de llamadas, auto-completado de adaptación, construida en la documentación, los puntos de interrupción en tiempo real, soporte de pantalla dividida, vista previa de los activos y un simulador de distancia.

- **Corona complete** es un depurador visual, editor de código y director del proyecto de Corona SDK. La herramienta ofrece un auto-complete inteligente, variable de memory dumping, salida de consola, gestión de archivos, error y la gestión de punto de interrupción.

- **Spriteloq** es un Adobe Flash SWF exportador de Corona SDK que permite a los desarrolladores integrar el arte de Flash y animaciones en las aplicaciones móviles.

- **Lime** es una biblioteca de Lua creado para Corona SDK que permite a los desarrolladores de juegos incluir mapas de baldosas creadas en baldosa dentro de un juego. Baldosa puede ser utilizado como un editor de niveles y como una característica de propiedad que permite a los diseñadores de niveles para establecer la lógica del juego sin la necesidad de un programador para modificar el código.

- **SpriteHelper** es una aplicación de Mac OS X que trae a los desarrolladores una textura y un editor de forma física en un solo paquete.

- **Corona Textmate Bundle** está diseñado para ayudar a los usuarios Corona SDK con aplicaciones de código TextMate más rápidamente. Contiene un gran número de términos de auto-complete, comandos y snippets de acceso a varias API de Corona utilizando los TextMate keyboard/menu shortcuts

- **Particle Candy** es una biblioteca de motor de partículas que permite a los desarrolladores crear humo, senderos, fuego, explosiones, escombros, bengalas, el polvo, las nubes, los tiros, los rayos de luz, fuentes de agua, y los efectos del clima.

- **MultiRezer** es una utilidad para los desarrolladores de juegos de puertos y aplicaciones a múltiples dispositivos, ajustando la base de gráficos para ajustar un dispositivo deseado.

- **LevelHelper** es un editor de niveles para Mac OS que ayuda a poblar los mundos del juego. Se utiliza junto con SpriteHelper, los desarrolladores pueden importar sprites y crear niveles arrastrando sprites en una posición deseada.

- **Icon Robot** es una herramienta que genera diferentes iOS tamaño iconos de cambio de tamaño y exportación de iconos de aplicaciones

- **Corona Remote** es una aplicación y biblioteca de Lua diseñado para enviar de forma remota los datos del acelerómetro de Corona SDK. El uso de un iPhone 3G, iPhone 3GS, iPhone 4, iPod Touch o iPad, los desarrolladores pueden enviar los datos del acelerómetro en tiempo real al Simulador de Corona para su uso en el desarrollo de aplicaciones y juegos de potente acelerómetro.

- **Texture Packer** es una interfaz gráfica de usuario que ofrece soporte multi-touch, zoom y una vista de árbol con sprites y muestra los cambios en tiempo real.

- **Physics Editor** ayuda a los desarrolladores en las formas de edición de física. Para arrastrar formas dentro del editor y presionando el botón automático, los desarrolladores pueden ajustar los parámetros de la forma de la física y directamente exportar a un marco de desarrollo de juegos.

- **iAppHost** crea sitios web de aplicaciones para iPhone con un CMS personalizado que da cuenta de HTML, CSS y análisis.

- **Outlaw** (antes Corona Project Manager) gestiona proyectos y activos que los desarrolladores puedan pasar de una aplicación a otra e intercambiar activos.

- **700 Sprites** es muy útil, ya que proporciona una amplia gama de temas de fantasía para el desarrollo de aplicaciones.

- **bmGlyph** es un generador de fuente de mapa de bits. Amplia gama de estilos están disponibles como degradados, brillante, sombras, etc.

- **Explosion Generator 3** utiliza la mecánica de motores avanzados para crear efectos explosivos perfectos de una manera sencilla. Esto puede ser un activo importante para el desarrollo del juego.

- **Game Coder** es una aplicación que depende de la plataforma que se puede utilizar para desarrollar aplicaciones en iOS. Ofrece la opción de exportar el código en Corona SDK y se pueden editar. Y éste puede se publicará en la App Store.

- **Glitch Game Libraries** son un conjunto de bibliotecas que se pueden utilizar para acelerar el tiempo de desarrollo.

- **Glyph Designer** es un personalizador de fuente que ayuda a añadir efectos a cualquier tipo de fuentes. Glyph Designer también tiene soporte de línea de comandos que permite el proceso de generar múltiples fuentes en diferentes tamaños para ser añadido a su proceso de construcción automatizado.

- **Gymbyl.com** es una página web que proporciona información y código para desarrollar, probar y mejoramiento integral.

- **Krea IDE** es una herramienta utilizada en la plataforma Windows que simplifica el uso de Corona SDK. Incluye muchas características como editor de escenas, editor cuerpo físico, el editor de código, editor de mapas, gestión de activos y muchos más.

- **Kutt** es una herramienta que puede ayudar en el desarrollo de la aplicación para diferentes resoluciones de pantalla. Esta herramienta hace el trabajo

más sencillo, ya que los códigos de la aplicación son para diferentes pantallas.

- **Level Director** es una herramienta de creación de nivel escrita específicamente para su uso con Corona SDK y Windows OS. El conjunto de herramientas proporcionado le permite crear rápidamente pantallas de títulos y niveles con facilidad y exportarlos a código LUA.

- **Million Tile Engine** es una biblioteca de LUA que brinda Multiple Tile Engine que está ardiendo rápido. Ayuda en el desarrollo de múltiples azulejos con independencia del tamaño de la pantalla y ejecutarlos sin tener en cuenta el tamaño del mapa.

- **MovieClipX** es una herramienta que se utiliza para hacer animaciones a un ritmo más rápido. Se prevé la posibilidad de proporcionar a los personajes de las animaciones y el estado del juego.

- **OPTPiX SpriteStudio** es una herramienta de creación de datos 2D animación sprite. SpriteStudio exporta datos de la animación en XML "Universal" o formatos binarios y actualmente soporta varios marcos y motores de juego, incluyendo Corona SDK.

- **Roaming Gamer Templates and SSKCorona** es una guía que proporciona todas las plantillas para todos los juegos y aplicaciones.

- **Spine** reemplaza animación de trama tradicional en los juegos, proporcionando con animaciones más suaves que son más fáciles producir. Las animaciones se pueden crear sin necesidad de más arte y son tan pequeñas que los juegos pueden hacer uso extensivo de ellos. Oficialmente es compatible con Corona SDK.

- **tPacker** genera atlas de texturas y animaciones de los sprites. También crea los iconos de las aplicaciones tanto para iOS y Android en el corto espacio de tiempo.

- **ZeroBrane Studio** es un IDE para Lua y los marcos basados en Lua que soporta resaltado de sintaxis, analizador de código, la consola remota, la depuración en el dispositivo y la codificación en directo.

6.5.6 Desventajas

Debido a que ANSCA no es oficialmente parte ni de Apple ni de Android, hay ciertas cosas que pueden no estar disponibles en la última versión del SDK nativo. Sin embargo, la gente que se encuentra trabajando en Corona está agregando características de forma constante a medida que van siendo disponibles.

6.6 TECNOLOGÍAS MULTIPLATAFORMA

6.6.1 PhoneGap.

PhoneGap es un framework de desarrollo de aplicaciones para dispositivos móviles multiplataforma. Permite crear aplicaciones con aspecto nativo que se pueden desarrollar en diferentes plataformas móviles: iOS, Android, Blackberry, Windows Phone, Web Os, Symbiam y Bada.

La gran ventaja de este framework es la posibilidad de desarrollar estas aplicaciones multiplataforma con un único código base utilizando las tecnologías web: HTML5, CSS3 y JavaScript.

Phonegap trata de solventar el problema de decidir la plataforma con la que trabajar cuando se comienza a crear una aplicación móvil. Que además también supone elegir el lenguaje de programación, ya que cada plataforma utiliza un lenguaje.

En un principio fue desarrollado por Nitobi y su uso era gratuito. En 2011 Adobe anunciaba la adquisición de Nitobi, por lo tanto PhoneGap pasaba al control de Adobe. De modo que ha sido integrado en las últimas versiones de Dreamweaver. Se armó un gran revuelo por el temor de que se abandonara la gratuidad, pero el código fue entregado a la Fundación Apache y así PhoneGap continuaba siendo un software libre. Actualmente este proyecto en la Fundación Apache recibe el nombre de “Apache Cordova” aunque se sigue manteniendo PhoneGap como una especie de marca comercial y por ello se sigue conociendo al framework como PhoneGap.

Dependiendo de la plataforma se usa un sistema distinto. En el caso de Android se utiliza Eclipse, disponible en Windows, Mac y Linux y una plantilla específica proporcionada por PhoneGap. Para iOS se usa Xcode, que solo está disponible en Mac y otra plantilla específica. Para el desarrollo en la plataforma de Blackberry no hay un entorno específico, se usa Java SDK y Blackberry SDK.

También es necesario instalar los SDKs de cada plataforma. Siendo SDK el conjunto de herramientas, librerías y compiladores que permiten desarrollar aplicaciones en un sistema concreto.

PhoneGap no se trata de un IDE, es decir, un entorno de programación como Eclipse o Xcode, no dispone de constructor de interfaz gráfica, ni de editor de código, tampoco de simulador.

Es posible encontrarlo en formato de plugin para utilizarlo en varios programas, como por ejemplo para Eclipse o como plantilla para Xcode en Mac. Y es compatible con frameworks de desarrollo web móvil como por ejemplo JQuery Mobile, Sencha Touch, jQToucho y muchos más.

Phonegap permite convertir casi todos los códigos basados en tecnologías web (HTML, CSS y JavaScript) en código con apariencia de nativo, preparado para compilar en el SDK. Evitando al programador el largo proceso de aprendizaje de lenguajes específicos de programación y facilitando el desarrollo y mantenimiento de la aplicación.

Con apariencia nativa nos referimos a que en realidad no se trata de una aplicación nativa. Es decir, PhoneGap empaqueta las aplicaciones web dentro de una aplicación nativa, de cualquiera de las plataformas soportadas (Android, iOs, Blackberry, Windows Phone, Web Os, Symbian), de modo que parece una aplicación nativa, pero no es así, se trata de una aplicación híbrida.

La interfaz de usuario con PhoneGap consiste en una vista de navegador web pero sin la barra de url. Ocupa el 100% del ancho y del alto del dispositivo. Se trata de la misma vista web que en el sistema nativo.

PhoneGap consiste en un conjunto de APIs (interfaz de programación de aplicaciones), basadas en JavaScript que permiten al desarrollador acceder a los elementos nativos del dispositivo, como la cámara, el acerelómetro, los contactos, el GPS, etc.

El uso de estas librerías se combina con un framework de desarrollo web móvil como puede ser jQuery Mobile, Dojo Mobile o Sencha Touch, de este modo es posible desarrollar aplicaciones que accedan a partes nativas del dispositivo utilizando únicamente HTML, CSS y JavaScript.

6.6.1.1 Ventajas de PhoneGap.

- Es gratuito.

- Es multiplataforma ya que corre dentro de un navegador web. Compatible con varias plataformas iOS, Android, Blackberry, Windows Phone, webOS, Symbian y Bada.

- Es fácil de desarrollar y ofrece muchas posibilidades a aquellos que conocen bien los lenguajes basados en tecnologías web (HTML, CSS y JavaScript).

- Existe mucha documentación acerca de PhoneGap e inluso la propia web proporciona muchos ejemplos.

- La alternativa que ofrece mediante el compilador en la nube es una forma muy sencilla que permite mejorar muchas de las desventajas de PhoneGap.

6.6.1.2 Desventajas de PhoneGap.

- En algunos casos es necesario usar el sistema operativo de la plataforma.

  Por ejemplo empaquetar aplicaciones Windows Phone solo es posible con el sistema operativo Windows. Lo mismo ocurre con iOS, es necesario usar un Mac.

Dependiendo de la plataforma se necesita un sistema diferente, para Android se requiere el uso de Eclipse y para iOS el uso de Xcode.

- Es necesario el uso de frameworks de desarrollo web móviles como por ejemplo jQuery Mobile, Sencha Touch. En sí se trata de una aplicación web por lo que el aspecto de la aplicación depende del framework web utilizado.

- Se desarrolla una aplicación híbrida, por lo que el rendimiento no es como el de una aplicación nativa.

- En caso de compilar para iOS es necesario introducir un código de desarrollador dado por la empresa Apple.

6.6.2 Titanium Appcelerator.

Se trata de una plataforma creada por la empresa Appcelerator en el año 2008 y mediante la cual se han sido desarrolladas más de 50.000 aplicaciones.

Permite desarrollar aplicaciones nativas para varios dispositivos móviles como Android, iOS, Blackberry y Windows Phone. También permite desarrollar aplicaciones híbridas basadas en tecnologías web como JavaScript, HTML, CSS y PHP, Ruby y Phyton.

Al igual que PhoneGap, mediante el uso de lenguaje web, proporciona una alternativa a los lenguajes nativos. Pero la diferencia es que con Titanium se puede generar una aplicación nativa y no un empaquetado que se ejecute en el navegador.

Para programar proporciona un IDE (entorno de desarrollo integrado) basado en eclipse, Titanium Studio, mediante el cual se puede crear proyectos y editar archivos JavaScript u otros recursos. Las aplicaciones nativas resultantes funcionan igual que si fueran programadas con el lenguaje propio de cada plataforma como Java en el caso de Android u Objetive C en el caso de iOS.

Titanium facilita el desarrollo de aplicaciones móviles multiplataforma con un único código base, con el 60% y 90% de código reutilizable entre plataformas.

Permite este desarrollo de forma rápida, probar y subir las aplicaciones a los canales de compra de aplicaciones para poder distribuirlas.

Es posible instalar Titanium Studio en Windows, Mac y Linux, pero con algunas limitaciones. Las aplicaciones iOS solo es posible desarrollarlas en el sistema operativo de Apple ya que las herramientas necesarias como Xcode solo es posible instalarlas en su propio sistema operativo. Lo mismo ocurre con Windows Phone solo es posible su desarrollo en el sistema operativo de Windows.

Como se ha mencionado antes se mantiene un código base entre las distintas plataformas, simplemente es necesario añadir algunas llamadas a las APIs específicas de la plataforma para la que se diseña. Es decir, se desarrolla un código común al que posteriormente se le añaden funciones propias de la plataforma, de modo que se simplifica el desarrollo y el mantenimiento de aplicaciones móviles.

El desarrollo de la aplicación se hace íntegramente en JavaScript por lo que todos los controles tienen que ser creados manualmente y para ello cuenta con una librería JavaScript que permite acceder a los controles del sistema. Así todos los controles (botones, listas, menús) son nativos, lo que hace que sean más rápidos. Otra diferencia con PhoneGap es que no hay DOM, lo que hace que no sea posible utilizar librerías de jQuery, porque con Appcelerator se trabaja con JavaScript puro, no dentro de un documento HTML.

Para desarrollar iOS es necesario utilizar Xcode, un entorno de desarrollo oficial de Apple. Mediante Titanium Studio se genera un proyecto Xcode con el JavaScript transformado y todas las librerías utilizadas. Una vez creada la aplicación en Xcode es posible probarla en el simulador del propio Titanium Studio. El proyecto Xcode generado se puede abrir en Xcode para configurar algunos aspectos de la aplicación como el logo pero no se puede editar el JavaScript desde Xcode, para ello se debe utilizar Titanium Studio.

En el caso de la plataforma Android, para usar el simulador y empaquetar la aplicación solo es necesario el SDK de Android.

El lenguaje JavaScript utilizado para desarrollar la aplicación no se ejecuta en el navegador del dispositivo como ocurre con otros frameworks como jQuery o Sencha. El código se ejecuta en un motor JavaScript que tienen todos los dispositivos, en el caso de Android y Blackberry es Mozilla Rhino y en iOS JavaScriptCore.

De modo que el código JavaScript se traduce al lenguaje nativo que utiliza la plataforma para la que se está diseñando y posteriormente es compilado a código nativo. Así es posible diseñar aplicaciones más rápidas y eficientes que con otros frameworks de JavaScript.

Quizás algo que ha sido desarrollado para iOS no funciona en Android o viceversa. Las librerías JavaScript son diferentes de modo que hay que informarse de lo que es posible hacer con cada plataforma.

Se ha observado anteriormente como PhoneGap contaba con una librería JavaScript para acceder a los elementos del dispositivo. Sin embargo Appcelerator precisa también de librerías para manejar los controles nativos y su distribución en la pantalla, lo que complica el desarrollo.

6.6.2.1 Ventajas de Titanium Appcelerator.

- Desarrolla aplicaciones móviles multiplataforma (iOS, Android, Blackberry y Windows Phone). Y también de escritorio (Windows, Mac y Linux).

- El aspecto y los controles son nativos, por lo que se obtiene mejor rendimiento.

- Gratis, soporte de pago.

- Reutilización del 60-90% de código en varias plataformas.
- Reduce los costes del desarrollo.
- El proceso de desarrollo es más rápido que con lenguajes nativos.
- La comunidad está en constante crecimiento.

6.6.2.2 Desventajas de Titanium Appcelerator.

- Los componentes visuales y los controles se definen manualmente mediante JavaScript.
- Para empaquetar aplicaciones iOS es necesario usar un Mac con Xcode instalado.
- Las librerías JavaScript son diferentes dependiendo de la plataforma.
- Escasa compatibilidad entre los sistemas operativos.

6.6.3 ADOBE AIR

Anteriormente denominado Apollo, AIR está descrito por la compañía como una tecnología de tiempo de ejecución (runtime) multiplataforma mediante la que se permite que los desarrolladores puedan utilizar sus actuales conocimientos de desarrollo web para la creación de aplicaciones Internet que puedan ejecutarse en los equipos de escritorio. Actualmente disponible como versión beta, se espera su lanzamiento para finales de año.

Según opina Lee Brimelow, “[AIR] crea un nuevo tipo de aplicaciones que se sitúan

entre las traidiconales aplicaciones de escritorio y las aplicaciones Web. Abre un nuevo mundo para los desarrolladores Web, permitiendo que los desarrolladores web puedan entrar en el escritorio. Por ejemplo, esto permite coger una cuenta de correo electrónico como Yahoo Mail y tener una aplicación offline que permita tanto leer mensajes existentes como componer nuevos mensajes. Una de las principales ventajas de AIR es que es multiplataforma, de modo que no tienes que preocuparte si se trata de un Mac o de un Pc".

AIR es en cierto modo un "envoltorio" para Flash, según indica Gabor Vida, presidente de Teknision; pero sin estar acoplado a ningún sistema operativo en concreto, continúa diciendo. "Sólo tenemos que desplegar un archivo AIR y es exactamente el mismo archivo para PC y Mac", con soporte para Linux en el horizonte próximo, indica Vida.

No obstante, AIR no es tan potente como Windows Presentation Foundation de Microsoft o las tecnologías de desarrollo Cocoa de Apple, tal y como indica Brimelow. Con dichas tecnologías, los desarrolladores tienen un acceso más completo a los discos duros de los usuarios y también a las capacidades proporcionadas por el sistema operativo. Pero WPF sólo está disponible para Windows, indica Brimelow.

Según declaraciones de Kurt Suchomel, fundador del Vermont Flash User Group y desarrollador de Draftfcb (una agencia de publicidad que también desarrolla sitios Web), "resulta interesante saber que existe un posible producto similar a FlashJester, pero con más versatilidad". Sin embargo, también ha expresado su preocupación sobre la eficacia real de AIR sobre los diferentes sistemas operativos, preguntándose si habría posibles "aspectos críticos" en cada una de las plataformas soportadas. Se supone que AIR será idéntico tanto en el Mac como en el PC, indica Suchomel. "Sólo espero que realmente sea así de sencillo. Será interesante ver cuáles serán los fallos. Mi preocupación natural se centra en la curva de aprendizaje para desarrollar con Apollo, incluso en el caso de que no soporte todas las

tecnologías principales de la web [HTML, JavaScript, CSS, XHTML Flash]".

En effectiveUI, quienes han desarrollado la aplicación de compras eBay Desktop utilizando AIR, el presidente de la compañía ha declarado que AIR podría complementar o desplazar a Java en diferentes apartados. AIR podría utilizarse para la creación de interfaces gráficas (una limitación de Java), mientras que Java podría residir en la parte del servidor.

Las tecnologías Flex y Flash de Adobe forman parte del framework Apollo, o AIR; pero AIR se diferencia de la nueva tecnología Silverlight de Microsof

Adobe AIR es una solución para la creación de aplicaciones en múltiples sistemas operativos que permite usar HTML/CSS, Ajax, Adobe Flash y Adobe Flex para ampliar las aplicaciones ricas de Internet (RIAs) al escritorio.

Las nuevas funciones de Adobe AIR incluyen una base de datos local integrada, soporte para PDF, capacidades mejoradas para desarrolladores JavaScript y una integración más profunda con Adobe Flex. La versión beta de Adobe AIR y el Software Developer's Kit (SDK) están disponibles como descargas gratuitas en el sitio web de la compañía.

6.7 SQlite

Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través

de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

El autor de SQLite ofrece formación, contratos de soporte técnico y características adicionales como compresión y cifrado.

6.7.1 Características

La biblioteca implementa la mayor parte del estándar SQL-92, incluyendo transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad (ACID), triggers y la mayor parte de las consultas complejas.

SQLite usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales. Por ejemplo, se puede insertar un string en una columna de tipo entero (a pesar de que SQLite tratará en primera instancia de convertir la cadena en un entero). Algunos usuarios consideran esto como una innovación que hace que la base de datos sea mucho más útil, sobre todo al ser utilizada desde un lenguaje de scripting de tipos dinámicos. Otros usuarios lo ven como un gran inconveniente, ya que la técnica no es portable a otras bases de datos SQL. SQLite no trataba de transformar los datos al tipo de la columna hasta la versión 3.

Varios procesos o hilos pueden acceder a la misma base de datos sin problemas. Varios accesos de lectura pueden ser servidos en paralelo. Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente.

En caso contrario, el acceso de escritura falla devolviendo un código de error (o puede automáticamente reintentarse hasta que expira un tiempo de expiración configurable). Esta situación de acceso concurrente podría cambiar cuando se está trabajando con tablas temporales. Sin embargo, podría producirse un interbloqueo debido al multihilo. Este punto fue tratado en la versión 3.3.4, desarrollada el 11 de febrero de 2006.

Existe un programa independiente de nombre SQLite que puede ser utilizado para consultar y gestionar los ficheros de base de datos SQLite. También sirve como ejemplo para la escritura de aplicaciones utilizando la biblioteca SQLite.

6.7.2 Lenguajes de programación

- La biblioteca puede ser usada desde programas en C/C++, aunque están disponibles enlaces para Tcl y muchos otros lenguajes de programación interpretado.

- SQLite se encuentra embebido en el REALbasic framework, haciendo posible que aplicaciones desarrolladas en REALbasic para Windows, Linux o Mac OS X usen la base de datos SQLite.

- Existe un módulo DBI/DBD para Perl disponible en CPAN, DBD::SQLite, no es una interface para SQLite, sino que incluye el motor completo de SQLite en sí mismo por lo cual no necesita ningún software adicional.

- Hay también un módulo para Python llamado PySQLite.

- Hay otro módulo para Visual Basic 6 llamado VBSqlite

- Desde Delphi se puede usar SQLite a través de los componentes libres ZeosLib.

- PHP incluye SQLite, desde la versión 5. SQLite también funciona con PHP 4 pero no viene incluido en él. Para más detalles vea el manual y PECL info.

- Desde Java se puede acceder mediante el driver de SQLite JDBC

- Desde .NET se puede acceder usando el proyecto de código abierto System.Data.SQLite

- Desde Lazarus 0.9.8 y Free Pascal 2.0.0, SQLite está disponibles para programadores de Pascal. Tutorial: «Lazarus Database Tutorial, Lazarus and SQLite» (en inglés).

- Mac OS X v10.4 incluye SQLite, y es una de las opciones en la Core Data API de Apple. AppleScript puede abrir, crear, y manipular base de datos SQLite por medio de la aplicación de ayuda "Database Events" de Mac OS X 10.4.

- BlitzMAX posee un MOD que permite trabajar con bases de datos SQLite. Para más detalles y descarga del MOD vea.

- El componente de base de datos (gb.db) de Gambas soporta SQLite en sus versiones 1, 2 y 3

- El lenguaje de programación de vídeo juegos Bennu tiene un mod de SQlite disponible

- El lenguaje de programación de scripting para Windows AutoIt v.3.x a traves de la DLL SQLite.dll.

6.7.3 Software que utiliza SQLite

SQLite es utilizado en una gran variedad de aplicaciones, destacando las siguientes:

- Adobe Photoshop Elements utiliza SQLite como motor de base de datos en su última versión del producto (la 6.0) en sustitución del Microsoft Access, utilizado en las versiones anteriores.4

- Clementine usa SQLite para guardar su colección de datos por defecto.

- Kexi usa SQLite como un motor de base de datos interno por defecto.

- Mozilla Firefox usa SQLite para almacenar, entre otros, las cookies, los favoritos, el historial y las direcciones de red válidas.4

- Los desarrolladores de OpenOffice.org han considerado incluir SQLite en el modelo de base de datos de Base, pero esto depende en gran manera del progreso de sqlite-sdbc-driver, que está todavía en estado de alpha. Actualmente han decidido usar HSQLDB.

- Varias aplicaciones de Apple utilizan SQLite, incluyendo Apple Mail y el gestor de RSS que se distribuye con Mac OS X. El software Aperture de Apple guarda la información de las imágenes en una base de datos SQLite, utilizando la API Core Data.4

- El navegador web Opera usa SQLite para la gestión de bases de datos WebSQL.

- Skype es otra aplicación de gran despliegue que utiliza SQLite.5 4

- SQLFilter, un plugin para OmniPeek, usa SQLite para indexar paquetes en una base de datos para poder ser consultada por medio de SQL.

- The New Yorker guarda el índice para un set de DVD conteniendo todos los números publicados por la revista.

- XBMC Media Center (antes conocido como "XBox Media Center") es un reproductor de medios de audio, video, fotos, etc de código libre (open source) multi-plataforma a la vez que un centro de entretenimiento. Usa SQLite para administrar las librerías de música, video y fotografías, listas de reproducción y bookmarks entre otras utilidades menores.

Debido a su pequeño tamaño, SQLite es muy adecuado para los sistemas integrados, y también está incluido en:

- Android
- BlackBerry
- Windows Phone 8
- Google Chrome
- iOS
- Maemo
- MeeGo
- Symbian OS
- webOS

6.8 PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

6.8.1 Visión general

PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Mediante extensiones es también posible la generación de archivos PDF,Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tanto SQL como NoSQL tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite o MongoDB.9

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java, CGI/Perl y a Node.js/Javascript. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio. CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno de desarrollo integrado para PHP, denominado 'Delphi for PHP. También existen al menos un par de módulos para Eclipse, uno de los entornos más populares.

6.8.2 Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

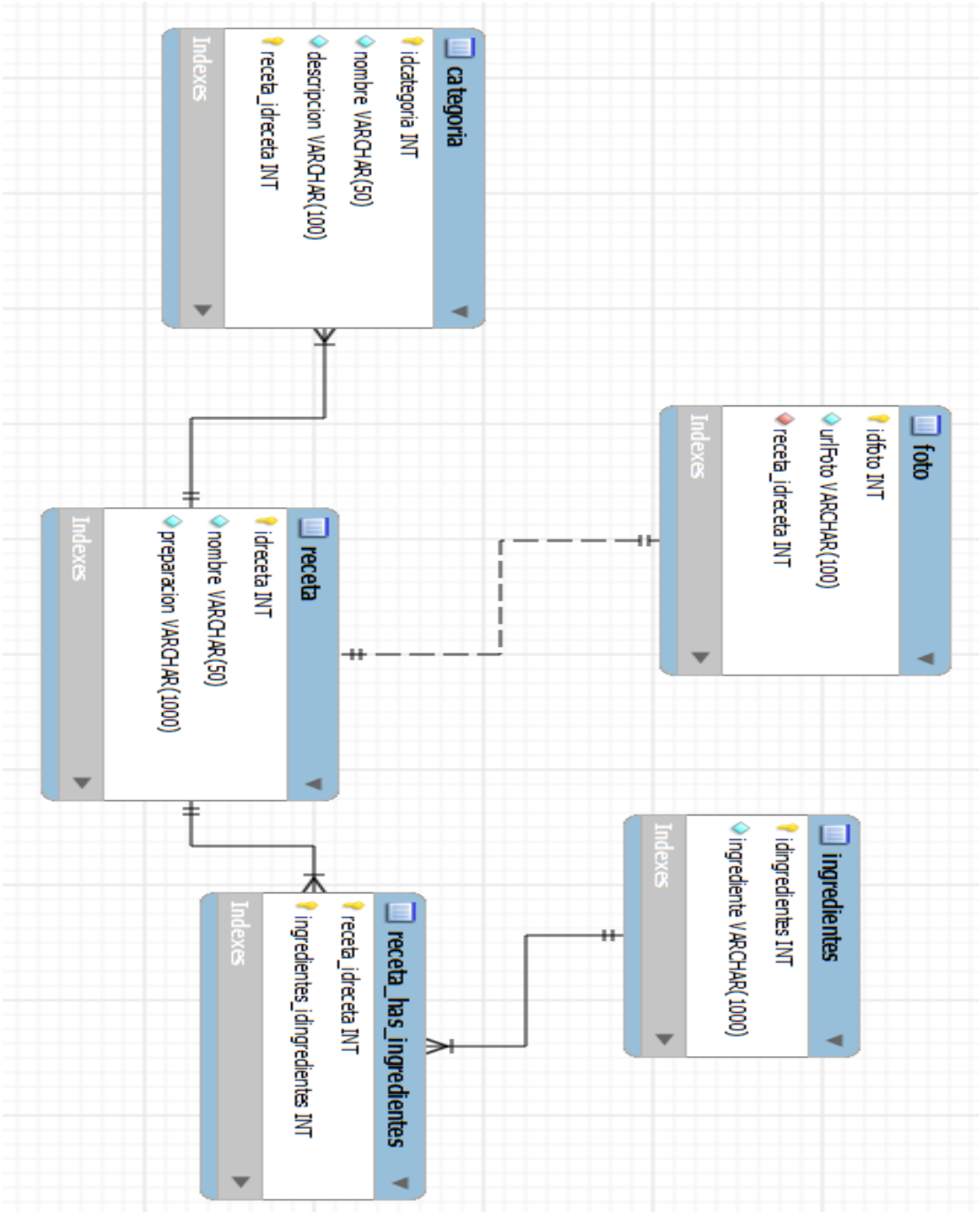
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.

- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

7. ANALÍSIS Y DISEÑO

7.1 Diagrama Entidad-Relación



7.2 Diseño de Datos.

Tabla 1: Categoría

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idcategoria	int		Código con el que identifica la categoría
nombre	varchar	50	Nombre de la categoría
descripcion	varchar	100	Descripción breve de la lista de recetas de dicha categoría
receta_idreceta	Int		Código con el que identifica una receta.

Tabla 2: receta

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idreceta	int		Código con el que identifica una receta
nombre	varchar	50	Nombre de la categoría
preparacion	varchar	1000	Detalla la preparación que se debe llevar para realizar una comida especifica

Tabla 3: ingredientes

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idingrediente	int		Código con el que identifica los ingredientes.
ingrediente	varchar	1000	Detalla todos los ingredientes necesarios para una comida.

Tabla 4: foto

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idfoto	int		Código con el que identifica la foto
urlFoto	varchar	50	Dirección en la que se ubica la imagen.
receta_idreceta	Int		Código con el que identifica una receta.

Tabla 5: receta_has_ingrediente

Nombre del Campo	Tipo de Datos	Longitud	Descripción
receta_idreceta	Int		Código con el que identifica una receta.
Ingredientes_idingrediente	int		Código con el que identifica los ingredientes.

8. CONCLUSIÓN

La elección de la herramienta o framework a utilizar, depende de las necesidades que la aplicación debe de suplir al tener en cuenta aspectos como el tipo de la aplicación a desarrollar. La incompatibilidad de plataformas es un problema que afecta a los desarrolladores de aplicaciones móviles ya que con los miles de tipos de dispositivos móviles que hay en el mercado debe de ser difícil la toma de decisión puesto que la orientación es generalmente a desarrollar la aplicación solo al dispositivo y a la plataforma específica, es decir que pueden estar obligados a desarrollar la misma aplicación varias veces en diferentes plataformas.

Para solucionar estos problemas utilizamos los frameworks de desarrollo de aplicaciones móviles multiplataforma. Con esta herramienta desarrollamos la aplicación en un lenguaje de programación diferente al que se utilizará nativamente en las diferentes plataformas.

La multiplataforma al igual que las demás tecnologías de desarrollo tiene sus pro y sus contras, las ventajas superan a sus debilidades al menos en el entorno de desarrollo corona SDK que abarata la creación de aplicaciones, reduce considerablemente el tiempo de programación y el número de programadores siendo así una gran opción para empresas desarrolladoras o personas que quieran incursionar al mundo de las App móviles, abarcando una cuota de mercado mayor al 95% de dispositivos a nivel mundial sin perder funcionalidad rendimiento y desempeño.

9. RECOMENDACIONES

Para el análisis del entorno en el que se desarrollara la aplicación y la funcionalidad de la misma se de tomar un debido tiempo para decidir bajo que plataforma es viable trabajar aprovechando al máximo los recursos del dispositivo móvil y las características más fuertes del entorno de desarrollo.

Incitamos a los docentes del Departamento de Computación de la UNAN-León, que promuevan e implementen las nuevas tecnologías que nacen día a día para desarrollar de App tal como es Corona SDK que es muy factible y hace que la enseñanza que brinden sea de primera colocándolos a la vanguardia.

A los estudiantes del Departamento de Computación de la UNAN-León, que indaguen más sobre estas nuevas tecnologías que se nos brinda hoy en día para el desarrollo de App que son unos de los mercados más amplios en el mundo tecnológico.

En general invitamos a todos aquellos aficionados de la tecnología que les gusta desarrollar que se atrevan a explotar todos los beneficios que Corona nos brinda con el hecho de crear App con un solo lenguaje de programación para varias compañías que son cabeceras en el mercado tecnológico.

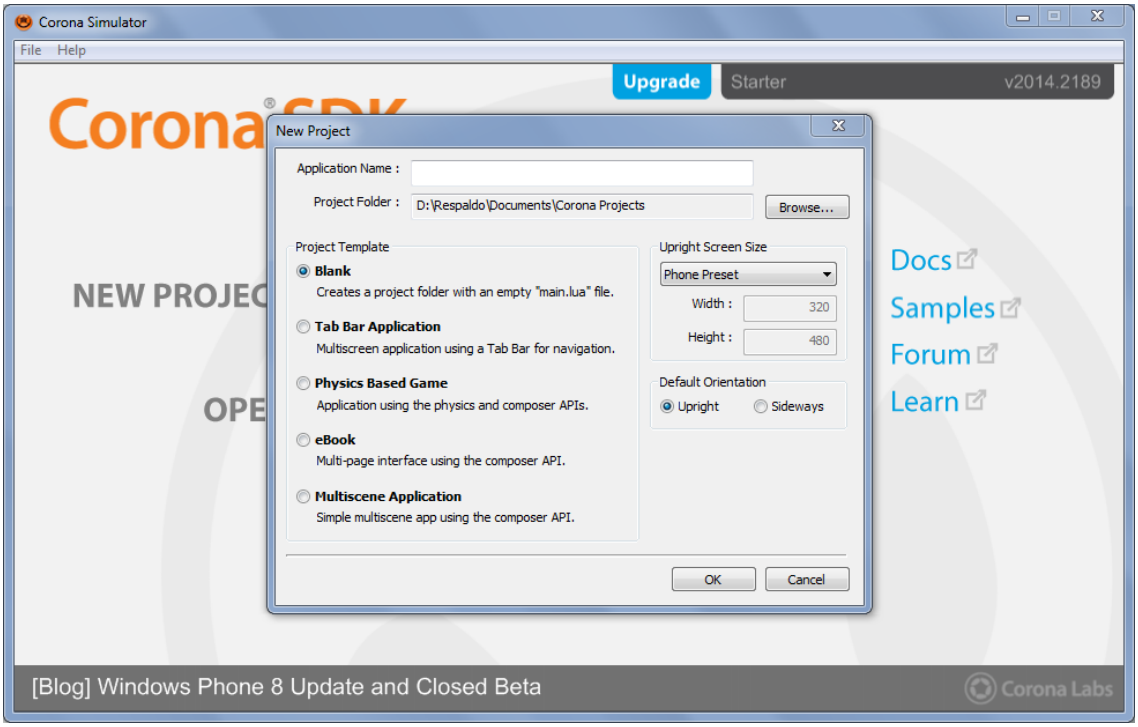
10. BIBLIOGRAFÍA

- http://www.coronalabs.com/
- http://www.coronalabs.com/products/corona-sdk/
- http://www.genbetadev.com/desarrollo-aplicaciones-moviles/corona-sdk-una-herramienta-de-desarrollo-para-ios-android-y-kindle-fire
- http://en.wikipedia.org/wiki/Corona_(software_development_kit)
- http://androideity.com/2011/08/24/corona-sdk-alternativa-para-desarrollar-juegos-android/
- http://learningcorona.com/
- http://mobile.tutsplus.com/tutorials/corona/corona-sdk-build-a-monkey-defender/
- http://www.coronalabs.com/resources/tutorials/getting-started-with-corona/
- http://www.taringa.net/posts/celulares/13710188/Tutorial-De-Corona-SDK-Entorno-Grafico-Tutorial-En-Video.html
- http://es.wikipedia.org/wiki/Multiplataforma
- http://www.developereconomics.com

11. ANEXOS

11.1 Entorno Corona SDK

Esta imagen muestra el entorno Corona SDK; en el podemos emular los proyectos creados de una determinada aplicación, este es el menú principal en que se encuentran varias opciones tales como las de crear un nuevo proyecto o la de abrir uno en desarrollo, es un entorno muy simple y fácil de comprender con el cual los usuarios podrán familiarizarse muy rápido.


Corona Simulator
File Help
Upgrade
Starter
v2014.2189
Corona SDK
NEW PROJEC
OPE
Docs
Samples
Forum
Learn
New Project
Application Name :
Project Folder :
D:\Respaldo\Documents\Corona Projects
Browse...
Project Template
Blank
Creates a project folder with an empty "main.lua" file.
Tab Bar Application
Multiscreen application using a Tab Bar for navigation.
Physics Based Game
Application using the physics and composer APIs.
eBook
Multi-page interface using the composer API.
Multiscene Application
Simple multiscene app using the composer API.
Upright Screen Size
Phone Preset
Width :
320
Height :
480
Default Orientation
Upright
Sideways
OK
Cancel
[Blog] Windows Phone 8 Update and Closed Beta
Corona Labs

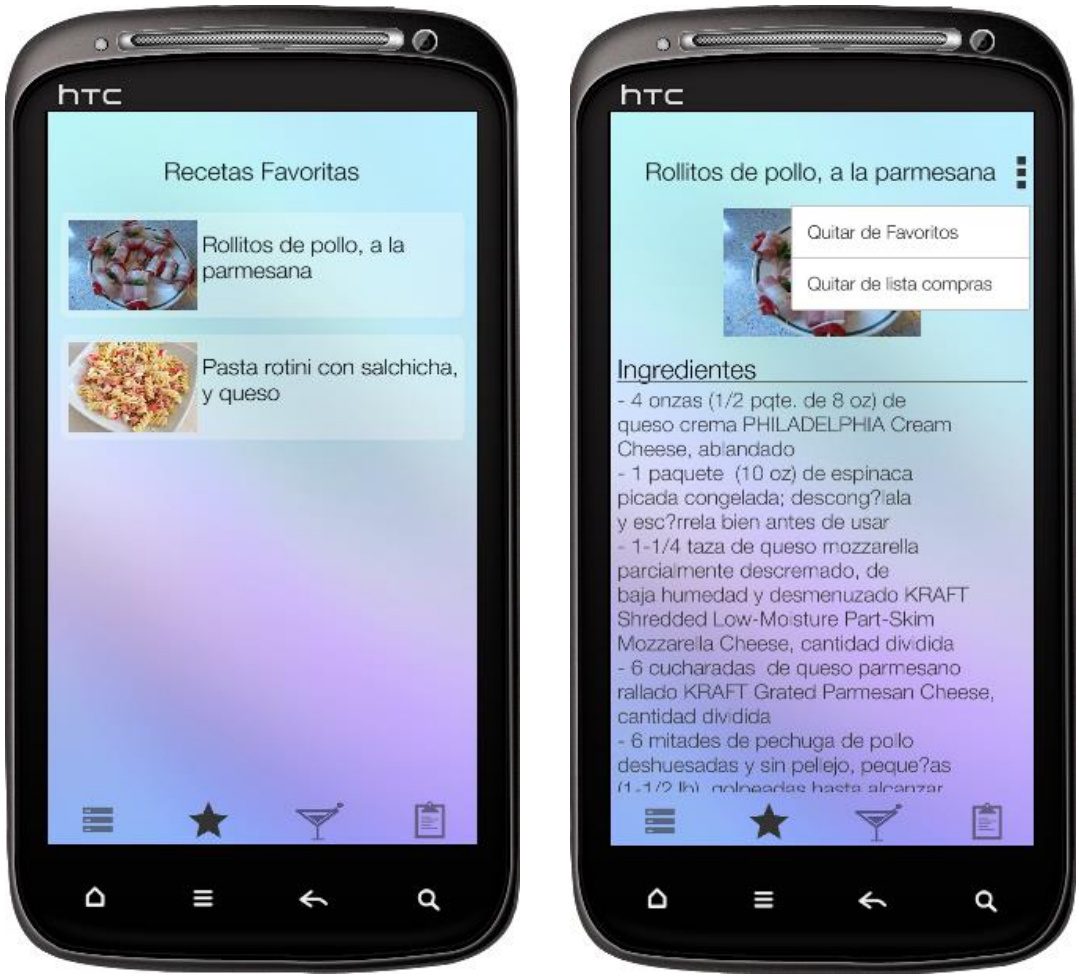
11.2 Aplicación de recetas de cocina desarrollada en Corona SDK

Para demostrar las opciones multiplataforma que posee Corona, las funciones de la App serán mostradas en diferentes tipos de dispositivos que abarca.

Al momento de iniciar la aplicación se nos mostrara la primera pantalla con un splash que en su mayoría de las aplicaciones y juegos tienen hoy en día. Esta es solo la presentación de la aplicación.

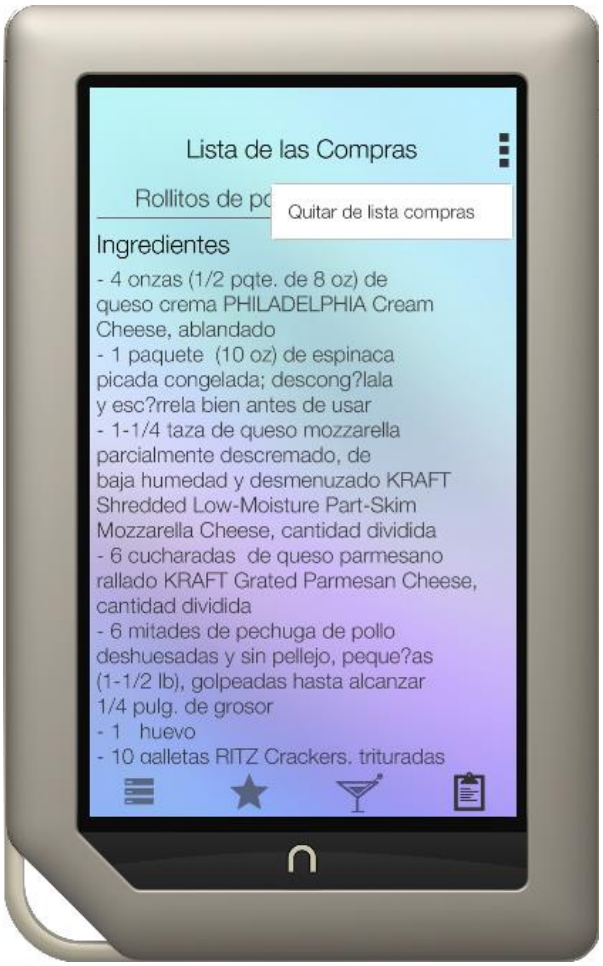
Al terminar la presentación automáticamente cargará una lista de categorías de comidas (solo si tiene conexión a internet) en la cual el usuario seleccionara la de su preferencia. Cabe resaltar que en la parte inferior se encuentra las diferentes opciones del menú que brinda la App tales como Categorías de Recetas, Favoritos, Bebidas, Lista ingredientes.

Al momento de seleccionar una categoría, este mostrara otra lista con las recetas de comida pero de una forma más vistosa para que el usuario se sienta atrapado al momento de ver la receta. Una vez que le haya gustado una receta específica el usuario tendrá la facilidad de añadirla a sus Favoritos o bien guardar los ingredientes de esta. Esto lo podrá hacer desplegando el menú de opciones que se encuentra en la parte superior izquierda de la pantalla que contiene Añadir a Favorito y Agregar a Lista de Compras o en su opuesto Quitar de favorito y Quitar de Lista de Compras. Esta función está siendo visualizada en móvil iPhone.



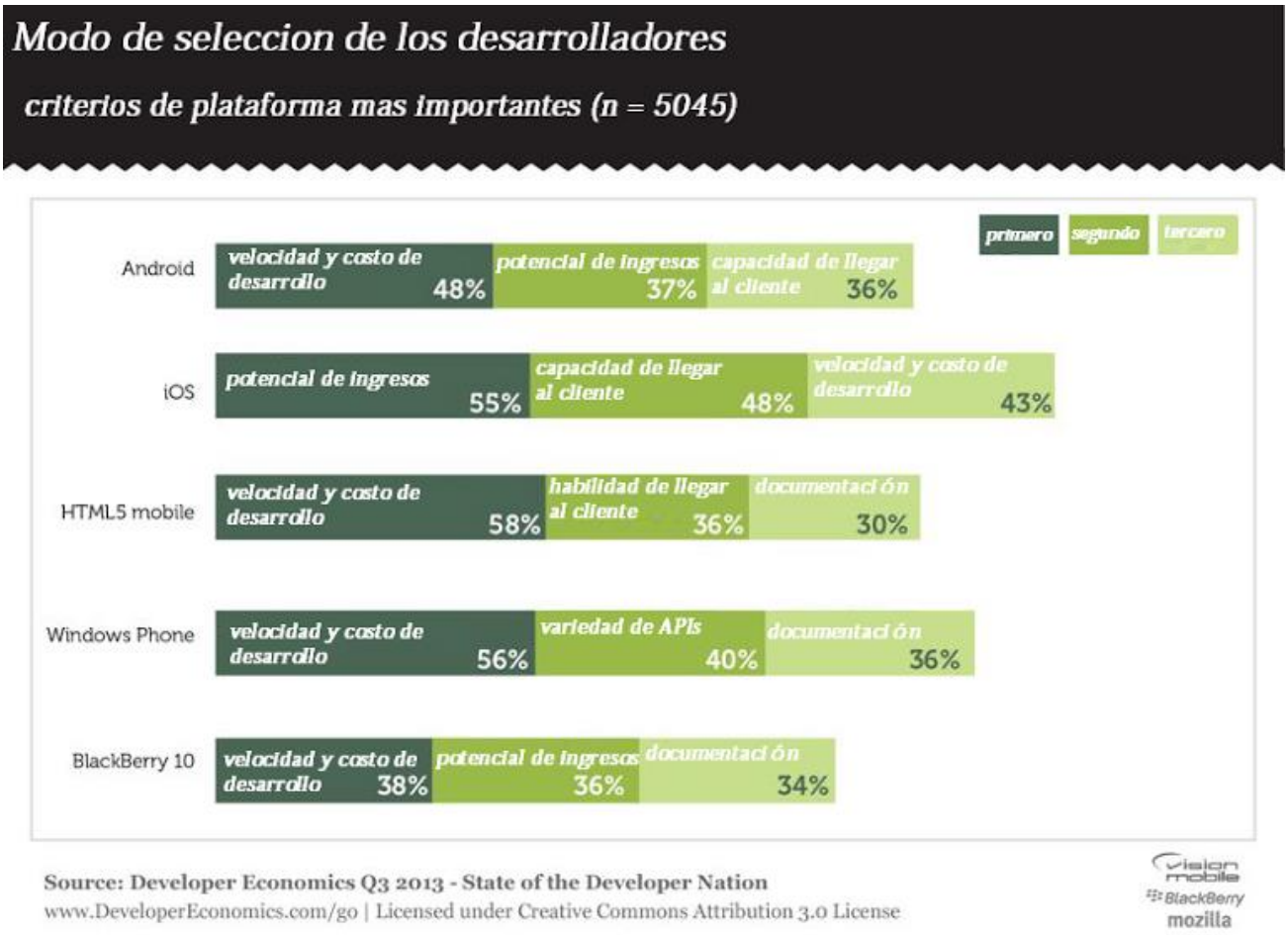
En la opción de favoritos el usuario podrá visualizar todas aquellas recetas que haya agregado en su búsqueda anterior. Cabe resaltar que podrán ser vista si necesidad de consumir internet ya que estas fueron debidamente almacenadas en la base de datos interna del móvil mediante SQlite. En caso que ya no desee tener esta receta guardada puede eliminarla muy fácil en el menú de opciones que podrá desplegar en su App ubicado en la parte superior izquierda de su pantalla, a su vez podrá también Agregar o Quitar de Lista de Compras. Esta función está siendo visualizada en móvil HTC con SO Android.

Opción de bebidas en la que se desplegara un alista con todas las bebidas que contenga, al seleccionar una de ellas se le mostrara los ingredientes y la preparación correspondiente a la bebida especificada. Con esta opción no podrá Añadir a Favoritos ni a Lista de Compras. Solo será visualizado con conexión a internet. Esta función está siendo visualizada en una Tablet Kindle Fire de Amazon con SO Android.



De última instancia esta la opción de Lista de compras. Para que esta tenga algún contenido tuvo que haber sido cargado anteriormente en la búsqueda del usuario o bien desde la lista de favoritos que tenga en su móvil. Estos datos podrán ser vistos sin necesidad de conexión a internet ya que están almacenados en la base de datos del móvil mediante SQLite. Esta solo podrá mostrar un alista de ingredientes de una recetas a la vez. Si selecciona otra receta automáticamente la que se encontraba almacenada será borrada y reemplazada por la nueva selección. O bien puede borrar esos datos de modo manual simplemente desplegando el menú en su pantalla que le dará la opción de Quitar de Lista de Compras. Esta función está siendo visualizada en una Tablet Nook Color con SO Android.

11.3 Selección de plataforma



11.4 Programación

En esta sección se mostrara fragmentos de código que son relevantes en el desarrollo de App.

```
---------------------- IMPORTAMOS LAS LIBRERIAS NECESARIAS ----------------------------
local widget = require("widget")
local storyboard = require("storyboard")
local json = require("json")
local scene = storyboard.newScene()
widget.setTheme( "widget_theme_android" )

------------------------------------ VARIABLES GLOBALES --------------------------------
local miFuente = "HelveticaNeue"
local pantAncho, pantAlto, pantAnchoMedio, pantAltoMedio = display.contentWidth, display.contentHeight,
display.contentCenterX, display.contentCenterY
local lista = nil local datosLista = {} local moverPantalla = false local servicioWeb = nil local
progresoWebService, totLisyta = 0, nil
local scrollView = nil local noDatos = false

============= FUNCION PARA CREAR LA LISTA ==================
function crearLista()
        lTop = 48 lLeft= 0 lWidth = 320 lHeight = 475
        if(system.getInfo("model") == "iPhone") then lTop = 43 end
        if(system.getInfo("model") == "iPhone" and display.pixelHeight < 970) then
                lHeight = 384
        elseif display.pixelHeight <= 800 then lHeight = 438 end
        lista = widget.newTableView{
                top = lTop, left= lLeft, width = lWidth, height = lHeight,
                maskFile = "images/mask-320x448.png",topPadding=20, hideBackground=true,
                onRowRender = onRowRenderLista, onRowTouch = onRowTouchLista,listener =
scrollListener,
            scrollBarOptions = { sheet = scrollBarSheet }
        }
        screengroup:insert( lista )
        llenarLista()
end

=============== FUNCION PARA LLENAR LA LISTA =============
local function llenarLista()
        rColor={ default =  { 0.847, 1, 1 }, over = { 0.847, 1, 1  } }  lColor={ 0.847, 1, 1 }
        for i = 1, #datosLista do
                lista:insertRow{
                        rowHeight = 72,  width=500,  listener = onRowTouchListbarras, rowColor = rColor--,
lineColor = lColor,
                }
        end
end
```

```
module(..., package.seeall)

--================= IMPORTAMOS LAS LIBRERIAS NECESARIAS ================--
require "sqlite3

--================== VARIABLES GLOBALES ==================--
local rutaBD = system.pathForFile("cocinaDB.db", system.DocumentsDirectory

========== FUNCION PARA CREAR LA BASE DE DATOS EN EL CEL ==========
function crearBD(  )
	print("\nCreado la base de datos si no exite")
	db = sqlite3.open( rutaBD )--ABRIMOS LA BD
	print("Creando la tabla recetasFavoritas si no existe")
	-- CREAMOS LA TABLA FAVORITOS --
	local tablaFavoritos = [[CREATE TABLE IF NOT EXISTS recetasFavoritas (id INTEGER PRIMARY
KEY, nombre, ingredientes, preparacion, urlFoto, descripcion);]]
	print("Creando la tabla de ingredientes si no existe")
	-- CREAMOS LA TABLA INGREDIENTES
	local tablaIngredientes = [[CREATE TABLE IF NOT EXISTS tablaIngredientes(id INTEGER
PRIMARY KEY, nombre, ingredientes);]]
	-- EJECUTAMOS LAS CONSULTAS --
	db:exec( tablaFavoritos )
	db:exec( tablaIngredientes )
	db:close() -- CERRAMOS LA CONEXION DE LA DB
end

========== FUNCION AÑADIR UNA RECETA A FAVORITOS O BORRARLA ==========
function anadirfavoritos( insertar, id, nombre, ingredientes, preparacion, descripcion, imagenUrl )
	-- body
	if(insertar) then
		--Guardamos la imagen en el DocumentDirectory
		db = sqlite3.open( rutaBD )--ABRIMOS LA DB
		print("Insertando Datos")
		local tablaInsert =[[INSERT INTO recetasFavoritas VALUES
('']]..id..[[',']]..nombre..[[',']]..ingredientes..[[',']]..preparacion..[[',']]..id..[[',']]..descripcion..[[');  ]]
		db:exec( tablaInsert )--INSERTAMOS LOS DATOS EN LA TABLA
		db:close()--CERRAMOS LA CONEXION
		print("\nanyadido a Lista de favoritos")
		local alert = native.showAlert( "Success", "Receta Añadida a Favoritos", { "Ok" } )
	else
		db = sqlite3.open( rutaBD )--BORRAMOS EL DATO
		print("Borrando Dato")
		local borrar =[[DELETE FROM recetasFavoritas WHERE urlFoto = ']]..id..[[';  ]]
		db:exec( borrar )--EJECUTAMOS LA CONSULTA
		db:close()--CERRAMOS LA CONEXION
		local alert = native.showAlert( "Success", "Receta eliminada de mis Favoritos", { "Ok"})
	end
end
```

```
============== EVENTO MANEJADOR DEL NETWORK.REQUEST ===============
function networkListener( event )
	-- body
	if(event.isError) then
		print("Network Error!!!") noDatos = true
		native.showAlert( "Warning", "Error de red, por favor revise sus ajustes de internet", { "OK"
} )
		spinner:stop() transition.to( spinner, { time=700, alpha=0} ) screengroup:insert(spinner)
miTexto.text = "Network Error!!!"
	else
		myServerData = event.response
		decodedData = (json.decode(myServerData))
		local counter = 1
		local listaCategoria = decodedData[counter]
				while (listaCategoria ~=nil) do
			datosLista[counter] = {}
			datosLista[counter].id = listaCategoria[1]
			datosLista[counter].title = listaCategoria[2]
			datosLista[counter].subtitle = listaCategoria[3]
			counter = counter + 1
			listaCategoria = decodedData[counter]
		end
		if(contador == 1) then spinner:stop() transition.to( spinner, { time=700, alpha=0} )
screengroup:insert(spinner)
			miTexto.text = "No hay datos Disponibles" noDatos = true
		else crearLista() spinner:stop() transition.to( spinner, { time=700, alpha=0} ) end
	end
end
=========== FUNCION PARA MANDAR A LLAMAR EL SERVICIO WEB ============
function llamarPHP(  )
	print("Cargando los Datos")
	transition.to( spinner, { time=700, alpha=1} ) local params={} params.timeout=10
screengroup:insert(spinner)
	servicioWeb =
network.request("http://173.236.60.250/~fiestani/cocina/allCategorias.php","GET",networkListener, params)
end
```