

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



ECOSISTEMA DE SERVICIOS MOVILES BASADOS EN FIREFOX OS

TRABAJO FIN DE MÁSTER

Noé Arturo Consuegra Marben

2013

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**ECOSISTEMA DE SERVICIOS MOVILES
BASADOS EN FIREFOX OS**

Autor
Noé Arturo Consuegra Marben

Director
Juan Carlos Yelmo

Departamento de Ingeniería de Sistemas Telemáticos

2013

Resumen

Los seres humanos siempre hemos tenido la necesidad de poder comunicarnos entre nosotros para poder sobrevivir, antes, los servicios de comunicación podían tardar mucho tiempo debido a que la tecnología no estaba tan desarrollada como lo podemos ver ahorita. Hoy en día el comunicarse es vital y tan común para todo tipo de relaciones interpersonales con la creciente llegada de los Smartphone y Tablets.

El teléfono inteligente se llama inteligente debido a su plataforma móvil o sistema operativo móvil. El mundo actual está muy familiarizado con este ecosistema moderno de las comunicaciones móviles, como por ejemplo Android, iOS, Windows Phone, Blackberry, Meego, etc. Dentro de este trabajo de Fin de Máster daremos un breve repaso sobre las tecnologías web que participan dentro de estos sistemas, discutiremos los ecosistemas de servicios enfocándonos en el plano comercial y de negocio de Apple/iOS, Android/Google, Firefox OS, entre otros.

Durante los últimos años se ha venido hablando sobre una supuesta batalla para ver quién ocupa el primer puesto dentro del mercado móvil actual, donde los competidores más fuertes sin duda alguna son Apple y Google y competidores que alguna vez fueron fuertes como Windows han quedado relativamente fuera de dicha batalla pero también han nacido competidores que podría entrar con mucha fuerza como el nuevo Firefox OS. Todo esto nos lleva a nuestro objetivo primordial, el analizar y estudiar dichos ecosistemas donde hablaremos sobre las ventajas y desventajas de cada uno y explicaremos dentro de cada ecosistema su modelo de negocio y desarrollo, su market-place, la plataforma en donde se ejecutan y en que dispositivos podemos encontrar dicho servicio.

Se revisaron algunos papers de organismos internacionales así como artículos en blogs, revistas digitales y foros para profundizar en el tema y así poder dar un análisis crítico y comparativo sobre el ecosistema de servicios móviles de cada producto. Al final, hablaremos sobre una propuesta de mejora y proponer nuevas ideas para trabajos a futuro.

Abstract

Human beings always had the need to communicate with each other to survive, many years ago some communication services could take many time to reach the target because the technology was not as developed as we see right now. Today communication is vital and so common for all types of relationships with the growing arrival of Smartphones and Tablets.

The smartphone is called smart because of its mobile platform or mobile operating system. The world today is very familiar with this modern ecosystem of mobile communications, such as Android, iOS, Windows Phone, Blackberry, Meego, etc. Within this Master Thesis work gives a short overview on web technologies involved in these systems, we discuss the business ecosystem of Apple/iOS, Android/Google, Firefox OS, etc .

In recent years there has been talk about a supposed battle to see who ranks first in the mobile market today, where the strongest competitors are undoubtedly Apple and Google and competitors that were once strong as Windows have been relatively out of this battle but also born competitors who could enter with great force as the new Firefox OS. All this leads to our primary objective, to analyze and study these ecosystems where we are going to talk about the advantages and disadvantages of each ecosystem and explain the business model and development, its market-place, the platform where they run and the devices where it is implemented.

We reviewed some papers from international organizations as well as articles in blogs, digital magazines and forums to pursue the subject and be able to give a critical and comparative analysis of the mobile services ecosystem of each product. Finally, we will discuss a proposal for improvement and propose new ideas for future work.

Índice general

Resumen	i
Abstract.....	iii
Índice general.....	iv
Índice de figuras.....	vii
Glosario.....	ix
1 Introducción.....	11
1.1 Contexto.....	11
1.2 Objetivos.....	13
1.2.1 Objetivo específico	13
1.3 Metodología.....	14
1.4 Estructura de la memoria.....	14
2 Estado del arte.....	15
2.1 Tecnologías Web.....	15
2.1.1 HTML5.....	15
2.2 JavaScript.....	21
2.3 Open Web APIs.....	25
2.3.1 API (Application Programming Interface)	25
2.3.2 Mashups.....	26
3 Ecosistema de Servicios.....	26
3.1 Ecosistema Biológico.....	30
3.2 La economía como un ecosistema.....	31
3.3 Ecosistema Digital de negocios. (DBE).....	31
3.4 Definición de Ecosistema de Negocios.....	32
4 El ecosistema de Apple/iOS.....	32

4.1	Modelo de negocio de Apple.....	33
4.2	Marketplace.....	36
4.3	Modelo de Desarrollo	37
4.3.1	Compilador Apple LLVM.....	38
4.3.2	Instruments	38
4.4	Objective-C.....	38
4.5	Plataforma de Ejecución.....	40
4.6	Dispositivos.....	41
5	El ecosistema de Android/Google.....	42
5.1	Modelo de Negocio.....	42
5.2	Marketplace.....	43
5.3	Modelo de Desarrollo.....	44
5.4	Plataforma de ejecución.....	45
5.5	Dispositivos.....	46
6	Los ecosistemas de RIM/Blackberry y Microsoft/Windows Phone.....	46
6.1	RIM/Blackberry.....	46
6.1.1	Modelo de Negocio.....	47
6.1.2	Marketplace.....	47
6.1.3	Modelo de Desarrollo.....	48
6.1.4	Plataforma de Ejecución.....	49
6.1.5	Dispositivos.....	50
6.2	Microsoft Windows Phone.....	50
6.2.1	Modelo de Negocio de Windows Phone.....	50
6.2.2	Marketplace.....	51
6.2.3	Modelo de desarrollo.....	51
6.2.4	Plataforma de ejecución.....	51
6.2.5	Dispositivos.....	52
7	El ecosistema de Firefox OS.....	53
7.1	Modelo de negocio.....	55
7.2	Marketplace.....	57
7.3	Modelo de Desarrollo.....	57

7.3.1	Estructura de las aplicaciones.....	58
7.4	Plataforma de ejecución.....	61
7.4.1	Arquitectura de Boot to Gecko	62
7.5	Dispositivos.....	65
8	Análisis comparativo de Firefox OS con otros ecosistemas.	66
8.1.1	Tabla comparativa entre los ecosistemas móviles.	69
9	Propuestas de mejora y Trabajo a Futuro.	70
10	Conclusiones	70
11	Bibliografía	72

Índice de figuras

Figura 1: HTML5 Canvas	18
Figura 2: Formatos de video y navegadores que lo soportan.....	19
Figura 3: Formatos de audio y navegadores que lo soportan.....	19
Figura 4: Ejemplo de programación en JavaScript	23
Figura 5: Representación en el navegador de la programación.	24
Figura 6: Fases de un Ecosistema de Negocios. [8]	27
Figura 7: Número de Aplicaciones en Google Play.....	43
Figura 8: Comparación de descargas entre Firefox e Internet Explorer.....	54
Figura 9: "Firefox Manifest"	59
Figura 10: Ejemplos de diseños para varios dispositivos.	60
Figura 11: Arquitectura B2G.....	62

Glosario

API - Application Programming Interface

B2B - Boot To Gecko

DDMS - Dalvik Debug Monitor Server

DOM - Document Object Model

HAL - Hardware Abstraction Layer

HATWG - Hipertext Aplication Technology Working Web Group

HTML - Hiper Text Markup Language

OS - Operating System

REST - Representational State Transfer

RIM - Research In Motion

ROA - Resource-Oriented Arquitecture

SDK - Software Development Kit

SOAP - Simple Object Access Protocol

W3C - World Wide Web Consortium

1 Introducción

1.1 Contexto

Desde tiempos remotos los seres humanos hemos tenido la necesidad de comunicarnos entre nosotros para poder convivir y relacionarnos, debido a eso la tecnología siempre va un paso adelante, para proporcionarnos las herramientas necesarias para una buena comunicación. El boom de la tecnología fue con la llegada de Internet y al mismo tiempo las tecnologías móviles dieron un gran salto.

Hoy en día, más de 40% de la población mundial está equipado con un teléfono móvil (esta tasa supera al 50% en la mayoría de los países industriales), y muchas personas consideran que el teléfono móvil es un accesorio esencial en su vida cotidiana. Sin embargo, los dispositivos que se ofrece en el mercado actual siguen siendo inadecuados para algunos de nosotros. En 2010, hubo dos conceptos esenciales en el área de TI, se trata de Green Computing, y otro es el sistema operativo del teléfono móvil.

Con el desarrollo del OS para el teléfono celular, los teléfonos inteligentes habían entrado en la corriente principal en los mercados desarrollados. En el tercer trimestre de 2012, la penetración de los envíos de teléfonos inteligentes superó 29% a nivel mundial, aunque esta cifra varía mucho de casi el 65% en los EE.UU. y más del 50% en Europa y 19% en Asia y el Pacífico, 17% en América Latina y 18% en África y Oriente Medio.

Los productos y servicios de los móviles, se basan en cuatro componentes clave para tener éxito [1], a saber:

1. Un modelo de negocio (facturación) con los patrones naturales de uso del servicio.
2. Los dispositivos deben ofrecer nuevas características que satisfagan las necesidades del cliente.
3. Aplicaciones a nivel de dispositivo que aprovechen las nuevas características.
4. El contenido diseñado para el consumo dentro de los escenarios de uso móvil.

Sin embargo, las ventas de teléfonos móviles no están fundadas en las características de hardware, pero si en la interfaz de usuario y las aplicaciones disponibles. Al igual que cualquier plataforma de teléfonos inteligentes, iOS y Android son impulsados por la economía de la demanda, donde la demanda generada tiene un efecto mucho más fuerte en las ventas.

Los desarrolladores de aplicaciones móviles se dirigen a nichos de mercado especializados que hacen uso de diferentes dispositivos móviles e inalámbricos en infraestructuras. Estas infraestructuras inalámbricas incluyen, entre otros, las redes de comunicación por satélite, redes de área local inalámbrica y la comunicación celular. En octubre de 2011, iOS y Android estaban a la vanguardia, con más de 500.000 y 300.000 solicitudes, respectivamente. El resto de las plataformas quedó atrás con menor demanda: BlackBerry cuenta con 35.000 solicitudes, 30.000 aplicaciones de Windows Mobile y Symbian 25.000 aplicaciones.

Las plataformas de desarrollo de aplicaciones son un conjunto de interfaces de programación de aplicaciones (API) que permiten acceder a cierto dispositivo móvil y opciones de red que utilizan las aplicaciones para funcionar en un dispositivo móvil en particular y/o de la red. Por lo tanto, las plataformas exitosas son un imán para la inversión financiera. Aplicaciones para plataformas como iOS y Android son capaces de atraer grandes inversiones financieras por parte de los desarrolladores, inversionistas y marcas. Tomando iOS como ejemplo, y la estimación de que una aplicación tiene un costo promedio de \$ 30.000 a desarrollar, las 500.000 aplicaciones de iOS representan una inversión promedio de \$15B en el ecosistema iOS. Esta inversión contribuye directamente a la rentabilidad de Apple y sus ventas de dispositivos estimando un \$71B a finales del año 2011 .

El 25 de julio de 2011, el Dr. Andreas Gal, Director de Investigación de la Mozilla Corporation, anunció el "Boot to Gecko" Proyecto sobre la lista de distribución mozilla.dev.platform. La propuesta de proyecto era el de "perseguir el objetivo de construir un sistema operativo independiente completo para la web abierta" a fin de "encontrar las diferencias que mantendrán a los desarrolladores Web para que puedan crear aplicaciones que son - en todos los sentidos - iguales a las aplicaciones nativas construidas para el iPhone [iOS], Android y WP7 [Windows Phone 7] ". El anuncio identificó

estas áreas de trabajo como una nueva Web API para exponer el dispositivo y las capacidades del sistema operativo, como el teléfono y la cámara, aplicaciones para probar estas capacidades, y el código de bajo nivel para arrancar en un dispositivo compatible con Android.

En julio de 2012, Boot to Gecko fue renombrado como "Firefox OS". En septiembre de 2012 los analistas de Strategy Analytics pronosticaron que Firefox OS tendría el 1% del mercado mundial de Smartphones en 2013 - su primer año de disponibilidad comercial.

1.2 Objetivos

Análisis del incipiente modelo de ecosistema de servicios móviles basados en Firefox OS y tecnologías web como HTML5 y JavaScript como alternativa a ecosistemas más asentados basados en aplicaciones nativas sobre sistemas operativos como Android o iOS con su propio modelo de negocio, marketplace, modelo de desarrollo, terminales, etc.

1.2.1 Objetivo específico

Elaboración de un estado del arte de tecnologías web como HTML5, JavaScript, Open Web APIs.

Análisis del estado del arte de ecosistemas de servicios móviles: Apple/iOS, Google/Android, RIM/Blackberry, Microsoft/Windows Phone.

- Modelo de negocio.
- Marketplace.
- Modelo de desarrollo.
- Plataforma de ejecución.
- Dispositivos.

Estudio y comprensión del nuevo Sistema Operativo Firefox OS y una comparativa con otros subsistemas.

Análisis crítico con propuestas de mejora y elaboración de conclusiones.

1.3 Metodología.

Para la realización de este trabajo de fin de Máster, se investigó en varios libros ofrecidos por la Biblioteca de la Universidad Politécnica de Madrid, también se investigó en varios artículos, foros, presentaciones, noticias etc. relacionados con el tema.

1.4 Estructura de la memoria.

Una vez dada la introducción y que los objetivos hayan quedado claros empezaremos hablando sobre las tecnologías web en el capítulo 2, dando un breve resumen sobre HTML5, JavaScript y las “Open Web APIs”

Después, en el capítulo 3 explicaremos lo que es un ecosistema de servicios y como nace este concepto dentro del mundo de los negocios, seguido de esto estudiaremos el ecosistema de Apple/iOS, hablando sobre su modelo de negocio, marketplace, modelo de desarrollo, plataforma de ejecución y dispositivos. En los capítulos 5 y 6 se explica el mismo ecosistema de servicios pero de Android, RIM/Blackberry y Windows Phone.

Dentro del capítulo 7 hablaremos sobre el ecosistema de Firefox OS, explicando cada uno de sus componentes para que en el capítulo 8 hagamos un análisis comparativo de Firefox OS con los demás ecosistemas.

Terminaremos este trabajo aportando algunas propuestas de mejora, trabajo a futuro y conclusiones en los capítulos 9 y 10 respectivamente.

2 Estado del arte.

2.1 Tecnologías Web

2.1.1 HTML5

HTML5 es la quinta versión del lenguaje básico de la World Wide Web, HTML. HTML5 funciona con dos variantes de sintaxis para HTML: la primera es la variante clásica de HTML, HTML5 y el segundo se conoce como XHTML. Es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Este nuevo lenguaje se encuentra todavía en el modo experimental, pero muchos programadores y desarrolladores de software ya han realizado demasiados programas que utilizan este lenguaje, debido a las mejoras y ventajas.

Hoy en día, algunos elementos en HTML son obsoletos, no utilizados o no utilizados de la manera que estaban destinados. Estos elementos se han eliminado o re-escrito en HTML5. Para manejar mejor el uso actual de Internet, HTML5 incluye también nuevos elementos para una mejor estructura, mejorar la manipulación, elaboración y contenido de los medios.

La evolución del lenguaje HTML en que se basa la web pronto podrían cambiar toda la industria de Internet móvil. La quinta versión de HTML se abre el camino a una nueva generación de aplicaciones y servicios en línea accesibles a todos los teléfonos con un navegador web móvil. Al poner en duda la necesidad de utilizar aplicaciones nativas específicamente diseñados para un sistema operativo para móviles, HTML5 también permitirán a los desarrolladores evitar el riesgo de fragmentación entre diferentes plataformas (iOS, Android, Windows Phone, Symbian, Meego, Bada y BlackBerry OS) o incluso dentro de una plataforma. Uno de los objetivos de los promotores de HTML5 es permitir a los desarrolladores concentrarse en la optimización de sitios web y servicios móviles en sí, independientemente de los sistemas operativos.

Tim Berners-Lee inventó HTML para el intercambio de documentos científicos y técnicos. Aunque 1993 se acredita como la fecha oficial de invención, que había, de hecho, estado haciendo experimentos con él desde 1980. Debido a que HTML creció rápidamente, pasando por tres versiones en un corto período de tiempo, se hizo bastante indócil y llena de etiquetas y

atributos de estilo. Los Navegadores no ayudaron, estos empezaron a tolerar todo tipo de marcas y de hacer todo lo posible para interpretar documentos ilegales (es decir, los documentos que no se ajusten a las normas de HTML). Para ayudar a frenar la indisciplina, HTML tuvo que aprender de XML, llamando así XHTML.

Dado que el lenguaje HTML se había centrado en la descripción de las páginas en lugar de en los mecanismos de gestión de contenidos, su inventor, Tim Berners-Lee quería añadir la dimensión "semántica" que le faltaba. Sin embargo, el nuevo XHTML lenguaje basado en las obras de la World Wide Web Consortium (W3C) no obtuvo el apoyo de los principales actores de la web. Apple, Google, editores del navegador Opera y la Fundación Mozilla (Firefox) se reunieron en el Hipertext Application Technology Working Web Group (HATWG) para desarrollar una tecnología alternativa. Este trabajo se reincorporó en las normas promovidas por el W3C en 2007. El W3C y WHATWG están cooperando para promover una versión unificada de estas tecnologías. Mientras está fragmentado ahora el mercado de terminales y aplicaciones móviles, el lenguaje HTML5 debería proporcionar las condiciones necesarias para el despliegue de una, web móvil abierta, a condición de que todos los fabricantes y distribuidores de sistemas operativos móviles incorporen las especificaciones de HTML5 en todos los navegadores.

Las aplicaciones web son criticadas por su falta de velocidad en comparación con las aplicaciones nativas y su incapacidad para hacer pleno uso de las funcionalidades de los teléfonos móviles. Por esta razón, HTML5 incorpora funcionalidades diseñadas específicamente para terminales móviles. Tienen el beneficio de un acceso simplificado a los recursos de hardware de los smartphones (geolocalización, cámara, brújula, sensor de movimiento, altavoces, etc.) servicios móviles y servicios multimedia, controlar gráficos vectoriales, vídeo, arrastrar y soltar, e incluso las operaciones off-line.

HTML5 también permitirá la integración de las nuevas tecnologías asociadas a las hojas de estilo (CSS3/CSS4). Esta evolución de la (Cascading Style Sheets) estándar CSS permiten el cambio de tamaño de las ventanas para hacer que las páginas sean legibles en las pantallas de diferentes tamaños.

El W3C también está trabajando para crear interfaces de programación (APIs) que permitirán a los navegadores a utilizar las funciones avanzadas de los móviles: un servicio web podría ser, por ejemplo, tener acceso a la libreta de direcciones o de la cámara sin tener que usar una aplicación nativa. Estos navegadores HTML5 soportaran "WebGL" para la visualización de objetos 3D, "IndexedDatabase" para el almacenamiento de datos en el navegador o la interfaz de "Archivo" para acceder a los archivos de los teléfonos inteligentes. Gracias a estas tecnologías, las aplicaciones basadas en navegador se convertirán muy rápido en aplicaciones nativas.

HTML5 también incorpora un sistema de etiquetas de vídeo que permite a los navegadores recientes transmitir secuencias sin utilizar plug-in, como en Flash, Aparte de la mera exhibición de los videos convencionales, la combinación de HTML5 con otras tecnologías abre el camino a nuevos usos para los vídeos integrados en los sitios, la rotación, la fragmentación o la integración de texto en el modo de flash. HTML5 también es un desarrollo crucial para plataformas de vídeo online. En consecuencia, hace varios meses YouTube desarrolló una versión Smartphone/HTML5 de su sitio. Videos en Dailymotion también están disponibles en formato HTML5.

APIs

HTML5 especifica secuencias de comandos de interfaces de programación de aplicaciones que se pueden utilizar con JavaScript. Las interfaces del modelo de objetos de documento (DOM) se extienden con nuevas características de facto. También hay nuevas API como por ejemplo: El elemento Canvas de dibujo 2D de modo inmediato, reproducción multimedia, aplicaciones web off-line, edición de documentos, arrastrar y soltar, manejo de historial del navegador, gestor de registro de tipo MIME, los datos micro, almacenamiento web, etc.

HTML5 Canvas

El elemento Canvas HTML5 se utiliza para dibujar gráficos, sobre la marcha, a través de secuencias de comandos (normalmente JavaScript). El elemento Canvas es sólo un contenedor para los gráficos. Se Debe utilizar una secuencia de comando para dibujar los gráficos. Canvas tiene varios métodos de dibujo, como por ejemplo, cajas, círculos, personajes, imágenes predeterminadas e inclusive se pueden introducir imágenes de tu elección.



Figura 1: HTML5 Canvas

Navegadores que soportan el elemento Canvas: Internet Explorer 9, Firefox, Opera, Chrome y Safari.

HTML5 Video

Ahora vamos a hablar de una nueva característica de HTML5 y esto es "Video en la Web". Hasta ahora, no ha habido ningún estándar para mostrar un video / película en una página web. Hoy en día, la mayoría de los vídeos se muestra a través de un plug-in (como flash). Sin embargo, los diferentes navegadores pueden tener diferentes plug-ins.

HTML5 define un nuevo elemento que especifica un método estándar para incrustar un vídeo / película en una página web.

Pero, ¿cómo funciona? El atributo de control añade controles de vídeo, como reproducción, pausa y volumen. También es una buena idea incluir siempre los atributos de anchura y altura. Si la altura y la anchura se establece, el espacio requerido para el vídeo está reservado cuando la página está cargada.

Sin embargo, sin estos atributos, el navegador no sabe el tamaño del vídeo y no puede reservar el espacio adecuado para ello. El efecto será que el diseño de la página cambiará durante la carga.

Browser	MP4	WebM	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	NO	YES	YES
Safari 5+	YES	NO	NO
Opera 10.6+	NO	YES	YES

Figura 2: Formatos de video y navegadores que lo soportan

- **MP4- MPEG 4 archivos con H264 video codec y AAC audio codec.**
- **WebM- WebM archives con VP8 video codec y Vorbis audio codec**
- **Ogg- Ogg archives con Theora video codec y Vorbis audio codec.**

El audio de HTML5 funciona exactamente igual que el "video", el único cambio es que se debe insertar el contenido del texto entre audio y las etiquetas de audio para los navegadores que no admitan la creación de etiquetas "audio".

El "audio" es un elemento que permite múltiples elementos de origen. Los elementos "Source" pueden vincular a los distintos archivos de audio. El navegador utiliza el primer formato reconocido. [2]

Browser	MP3	Wav	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	NO	YES	YES
Safari 5+	YES	YES	NO
Opera 10+	NO	YES	YES

Figura 3: Formatos de audio y navegadores que lo soportan.

Las especificaciones técnicas de HTML5 están todavía siendo modificados. Aunque es difícil llegar a un consenso entre los 329 miembros del W3C, cuyos intereses a menudo difieren en gran medida, los principales actores en el sector de Internet móvil ya han adoptado HTML5.

La adopción de HTML5 debería reducir el coste de desarrollo de aplicaciones móviles y cambiar el proceso de difusión de las aplicaciones móviles. Así, el openAppMkt y Jolicloud han lanzado aplicaciones diseñadas en HTML5 que no están disponibles en los "mercados" móviles, por lo tanto, ya no será necesario hacerle un "jailbreak" al teléfono móvil para instalar estas aplicaciones.

Las empresas que desarrollan servicios móviles están buscando desarrolladores capaces de manejar HTML5 y las tecnologías correspondientes.

Sin embargo, la adopción de HTML5 no significa el final de "aplicaciones nativas" y sus mercados. Las empresas continuarán adaptando y difundiendo sus servicios y contenidos para plataformas específicas, para segmentar sus ofertas (por reservar ciertos contenidos o funcionalidades de una plataforma u otra), para ganar la lealtad de sus usuarios o proponer las suscripciones o las formas más simples de facturación. En efecto, como Google ha señalado, la búsqueda de no amigables no se limita a aplicaciones en sí. En consecuencia, las dificultades para realizar los pagos que encuentran los usuarios de AndroidMarket pueden llegar a ser un obstáculo para el desarrollo de esta plataforma.

Si la opción de desarrollar sólo las aplicaciones nativas en una de las plataformas, está disponible para las empresas privadas, lo contrario es cierto para los usuarios públicos. De hecho, se trate de gobiernos o autoridades locales, la interoperabilidad entre los distintos Smartphone representa a la vez la garantía de una distribución más amplia de los usuarios y el ahorro para el desarrollo de servicios móviles para el público en general. Debe tenerse en cuenta que un gran número de usuarios públicos en los Estados Unidos están considerando el desarrollo de versiones "web móvil", para dar prioridad al desarrollo de sus servicios de telefonía móvil para el público en general. La madurez del lenguaje HTML5 también podría reforzar esta tendencia en los próximos meses.

2.2 JavaScript

Es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete utilizado para construir sitios web y poder hacerlos más interactivos. Fue creado por Brendan Eich de Netscape con el nombre de Mocha, para finalmente terminar como JavaScript. JavaScript es una marca registrada de Oracle Corporation. Comparte muchas características y estructuras del lenguaje Java, pero este fue creado de forma independiente y el mismo ha superado desde hace tiempo sus raíces de lenguaje scripting para convertirse en un lenguaje de propósito general robusto y eficiente. Es un lenguaje de programación de alto nivel, dinámico, sin tipo de interpretación que está bien equipado para ser de estilo orientado a objetos y con programación funcional [3].

Este lenguaje puede interactuar con el código HTML, permitiendo así a los programadores web utilizar contenido dinámico. La mayoría de las páginas Web usan JavaScript conjuntamente con todos los navegadores de la Web. Los dispositivos como consolas de juegos, computadores de escritorio, laptops, tabletas y teléfonos inteligentes usan intérpretes JavaScript, haciéndolo el lenguaje de programación más omnipresente de toda la historia. Todos los desarrolladores de Web necesitan saber del mismo para especificar el comportamiento de las páginas Web, conjuntamente con HTML para especificar el contenido de las páginas Web y CSS para especificar la presentación de las páginas Web. [4]

Muchas personas no están a favor de JavaScript debido a que tienen toda la vida programando en otros lenguajes de programación y por ende se reusan a aprender este lenguaje, el API del buscador (el DOM, Document Object Model) es un poco laborioso para trabajar en otro lenguaje debido a que está pobremente especificado y inconsistentemente implementado. Lo resaltante de JavaScript es la capacidad de que un programador que no tenga conocimientos del lenguaje pueda realizar el trabajo deseado, inclusive una persona con conocimientos básicos de programación en general puede trabajar en JavaScript. Esto es debido a que es un lenguaje con un enorme poder expresivo.

Este lenguaje tiene como base algunas ideas muy buenas y algunas muy malas. Las muy buenas ideas incluyen funciones, escritura suelta, objetos dinámicos, y una notación literal objeto expresivo. Las malas ideas incluyen un modelo de programación basado en variables globales.

JavaScript es el primer idioma Lambda en convertirse en una corriente principal. En el fondo, JavaScript tiene más en común con Lisp y Scheme que con Java. Es Lisp enmascarado en C. gracias a esto JavaScript un lenguaje muy poderoso. Debido a que la teoría es que la tipificación estricta permite un compilador para detectar una gran clase de errores en tiempo de compilación, la mayoría de los lenguajes de programación hoy en día exige la tipificación robusta. Cuanto antes podamos detectar y reparar errores, menos nos cuestan.

JavaScript es un lenguaje débilmente tipado, y debido a esto los compiladores de JavaScript son incapaces de detectar errores de tipo de escritura, lo que causa alarmante para las personas que vienen a JavaScript desde lenguajes fuertemente tipados. JavaScript tiene una muy poderosa notación literal de objetos. Los objetos pueden ser creados simplemente enumerando sus componentes. Esta anotación fue la inspiración para JSON, el formato de intercambio popular para datos. Un aspecto controvertido en JavaScript es herencia de prototipos. JavaScript tiene un sistema de objetos de clase libre en que los objetos heredan las propiedades directamente de otros objetos. Esto es muy poderoso, pero no es familiar para los programadores de formación clásica. En su mayor parte, sin embargo, esas opciones eran buenas, si inusuales. Pero había una opción que era particularmente mala: JavaScript depende de variables globales de vinculación. Todas las variables de alto nivel de todas las unidades de compilación son lanzados juntos en un espacio de nombres común que se llama el objeto global. Esta es una mala cosa porque las variables globales son malas, y en JavaScript son fundamentales. Afortunadamente JavaScript también nos da las herramientas para mitigar este problema.

La sintaxis del JavaScript es relativamente fácil de comprender y de poder emplear en el ámbito de programación. Lo primero en resaltar son las "Declaraciones", ya que estas son las series de instrucciones que necesita este u otro lenguaje de programación para poder construir un guión. Estas declaraciones deben estar escritas con una sintaxis clara para que esta forma pueda ser interpretada de forma correcta. Seguidamente podemos recalcar los "Comentarios", y estos se pueden emplear cuando se desea escribir algo para el beneficio propio. Para este caso el interpretador de JavaScript ignora estas declaraciones realizadas. Estos comentarios pueden ser de gran ayuda cuando se desea seguir el rastro del propio código, podemos compararlas con recordatorios o notas realizadas para saber que está aconteciendo en el guión. Las "Variables" que es todo lo que está sujeto a cambios, a las variables se le pueden asignar valores para poder ser declaradas o asignadas dentro del sistema. Para poder representar las variables se puede tener "Cadenas", que consisten en valores numéricos, letras, signos de puntuación, espacios, entre otros. También se pueden usar "Números", que difieren con los valores anteriores debido a que estos pueden ser decimales. También se pueden tener valores booleanos para representar datos de 2 posibles valores, por ejemplo cuando se requiere representar si una variable es verdadera o falsa. También se pueden tener variables de "Colección", que no es más que una asociación de varios valores a una sola variable, por ejemplo si declaramos la variable Madrid FC, cada jugador asociado al club deportivo formara parte de esa variable. Y este tipo de asociación se puede extender a un más, porque se pueden asociar otro tipo de valores como por ejemplo el número que representa el día cuando fue fundada la institución, etc. Se tienen también "Objetos" que son también un conjunto de valores asignados a una variable con ciertas propiedades específicas. Agregado a todo esto también se puede realizar ciertas operaciones con los mencionados tipos de variables, por ejemplo operaciones aritméticas, comparaciones, condiciones, y muchas otras que por la limitante del artículo no

se pueden detallar a fondo, Todas esas variables, funciones, condiciones cuando se unen para lograr lo requerido forman parte de la sintaxis del lenguaje JavaScript[5].

En la figura 4 se ve un ejemplo de cómo JavaScript puede mediante una etiqueta de guion (script) puede ser añadido a una página web, en el cual el comando de <script> le indica al navegador que el texto indicado sea representado como un guion, y con el comando </script> le indica que regrese al comando HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>The American Eggplant Society</title>
  </head>

  <body>
    <h1>The American Eggplant Society</h1>
    <p>Welcome to our site. Unfortunately, it is still
      under construction.</p>
    <p>We last worked on it on this date:
    <script type="text/javascript">
    <!-- Hide the script from old browsers
    document.write(document.lastModified);
    // Stop hiding the script -->
    </script>
    </p>
  </body>
</html>
```

Figura 4: Ejemplo de programación en JavaScript

En la figura 5 se denota como se ve representado lo programado en el navegador. [6]

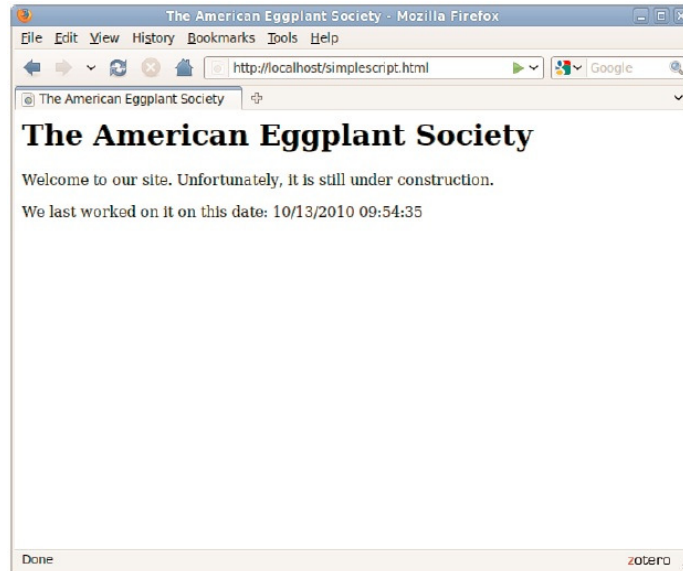


Figura 5: Representación en el navegador de la programación.

Habiendo descrito lo básico sobre JavaScript ahora trataremos de hablarle sobre su entorno en sistemas móviles que es la base fundamental de este artículo. La adopción generalizada de dispositivos móviles de gran alcance, particularmente aquellos con pantallas táctiles, ha requerido la creación de nuevas categorías de eventos. En muchos casos, los eventos de pantalla táctil se asignan a tipos de eventos tradicionales como el clic y desplazamiento.

Sin embargo, no todas las interacciones con una pantalla táctil de interfaz de usuario emula un ratón, y no todos los toques puede ser tratada como eventos de ratón. En el momento de escribir estas líneas, no hay estándares para estos eventos, pero el W3C ha comenzado a trabajar en un toque (Touch) de especificación de eventos que utilice eventos táctiles de Apple como punto de partida. Safari genera eventos de gestos para la ampliación de dos dedos y gestos de rotación.

El evento de gestión de entrada (`gesturestart`) se desencadena cuando comienza el gesto y el evento de gestión de salida (`gestureend`) se dispara cuando termina. Entre estos dos eventos es una secuencia de eventos cambio de gestión (`gesturechange`) que rastrean el proceso del gesto. El objeto del evento enviado con estos eventos tiene escala numérica y propiedades de rotación. La propiedad de la escala es la relación de la distancia actual entre los dos dedos a la distancia inicial entre los dedos. Un gesto "pellizco cerca" tiene una escala de menos de 1,0, y un "pellizco abierto" gesto tiene una escala mayor que 1,0. La propiedad de rotación es el ángulo de rotación del dedo desde el inicio del evento. Se informa en grados, con valores positivos que indican rotación en

sentido horario. Eventos de gesto son eventos de alto nivel que notifican al usuario de un gesto que ya ha sido interpretado. Se puede poner en práctica con sus propios gestos personalizados, puede detectar eventos táctiles de bajo nivel. Cuando un dedo toca la pantalla se activa un evento toque de comienzo (touchstart). Cuando el dedo se mueve, se activa un evento toque de movimiento (touchmove). Y cuando el dedo se levanta de la pantalla, se activa un evento de toque de finalización (touchEnd).

A diferencia de los eventos del ratón, los eventos de toque no reportan directamente las coordenadas del tacto. En su lugar, el objeto se envía con un evento táctil tiene una propiedad el cambio de toques (changedTouches). Esta propiedad es un objeto de matriz similar a cuyos elementos cada describir la posición de un toque. El evento de cambio de orientación (orientationchanged) se activa en el objeto Ventana del dispositivo que permiten al usuario rotar la pantalla de vertical a horizontal. El objeto que se pasa con un evento (orientationchanged) no es útil en sí. En Safari móvil, sin embargo, la orientación de la propiedad del objeto de ventana da la orientación actual como uno de los números 0, 90, 180, o -90. [4]

2.3 Open Web APIs.

2.3.1 API (Application Programming Interface)

Las APIs son interfaces de comunicación entre componentes de software. El principal propósito de una API consiste en proporcionar un conjunto de funciones de uso general, así los desarrolladores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

Una Open API son conjuntos de tecnologías que permiten a los sitios web que interactúen entre sí mediante el uso de REST (Representational State Transfer) SOAP (Simple Object Access Protocol), ROA (Resource-Oriented Architecture), JavaScript y otras tecnologías web.

Las Open Web APIs prácticamente es lo mismo que un servicio web ó lo que se le conoce hoy en día como la Web 2.0.

Las Open Web APIs permiten la combinación de varias APIs dentro de nuevas aplicaciones llamadas Mashups.

2.3.2 Mashups

Los Mashups simplemente son formas nuevas de crear nuevas aplicaciones web mediante la combinación de recursos web existentes que utilizan los datos y las API Web. Los Mashups son herramientas que tratan de compartir la información para apoyar la publicación de contenido para una nueva generación de aplicaciones Web. El objetivo de los mashups de servicios es el de diseñar y desarrollar nuevas y modernas aplicaciones Web que sean fáciles de alcanzar para el usuario final, combinando tecnologías de servicios web con nuevos contenidos, enfoques de colaboración (por ejemplo, tecnologías de la Web 2.0, las etiquetas y microformatos) y posiblemente, la gestión de datos y tecnologías de web semánticas. [7]

3 Ecosistema de Servicios.

Las empresas exitosas son aquellas que evolucionan con rapidez y eficacia. Deben atraer recursos de todo tipo, basándose en el capital, socios, proveedores y clientes para crear redes de cooperación.

En esencia, los ejecutivos deben desarrollar nuevas ideas y herramientas para la elaboración de estrategias, herramientas para la toma de decisiones difíciles cuando se trata de innovaciones, alianzas empresariales y de liderazgo de los clientes y proveedores.

Para extender un enfoque sistemático de la estrategia, una compañía no debe verse como el único miembro de la industria, sino como parte de un ecosistema de negocios que recorre una variedad de industrias. En un ecosistema de servicios, las empresas co-evolucionan capacidades en torno a una nueva innovación: trabajan cooperativamente y competitivamente para apoyar nuevos productos, satisfacer las necesidades del cliente, y, finalmente, incorporar la siguiente ronda de innovaciones.

Por ejemplo, Apple es el líder de un ecosistema que se compone de al menos cuatro grandes industrias: computadoras personales, dispositivos electrónicos para el usuario (iPod, iPad), la información y las telecomunicaciones (iPhone). El ecosistema de Apple incluye una red extensa de proveedores que incluye a Motorola y Sony y un gran número de clientes en diferentes segmentos del mercado.

El rol del líder es valorado por el resto de la comunidad. Este liderazgo permite a todos los miembros del ecosistema invertir para un futuro compartido en el que se prevé beneficios para todos.

Un ecosistema de negocios, al igual que su contraparte biológica, se mueve gradualmente a partir de una colección aleatoria de elementos para una comunidad más estructurada. Los Ecosistemas de negocios salen del remolino que origina el capital, los intereses de los clientes, y el talento generado por una nueva innovación.

Cada ecosistema empresarial se desarrolla en cuatro etapas distintas: el nacimiento, la expansión, el liderazgo y auto-renovación - o, si no, la muerte.

The Evolutionary Stages of a Business Ecosystem		
	Cooperative Challenges	Competitive Challenges
Birth	Work with customers and suppliers to define the new value proposition around a seed innovation.	Protect your ideas from others who might be working toward defining similar offers. Tie up critical lead customers, key suppliers, and important channels.
Expansion	Bring the new offer to a large market by working with suppliers and partners to scale up supply and to achieve maximum market coverage.	Defeat alternative implementations of similar ideas. Ensure that your approach is the market standard in its class through dominating key market segments.
Leadership	Provide a compelling vision for the future that encourages suppliers and customers to work together to continue improving the complete offer.	Maintain strong bargaining power in relation to other players in the ecosystem, including key customers and valued suppliers.
Self-Renewal	Work with innovators to bring new ideas to the existing ecosystem.	Maintain high barriers to entry to prevent innovators from building alternative ecosystems. Maintain high customer switching costs in order to buy time to incorporate new ideas into your own products and services.

Figura 6: Fases de un Ecosistema de Negocios. [8]

Durante la fase 1 de un ecosistema de negocios, los empresarios se centran en la definición de lo que los clientes quieren, es decir, el valor de un nuevo producto o servicio propuesto, y la mejor forma de su entrega.

La victoria en la etapa de nacimiento, en el corto plazo, va a menudo a los que mejor definen y ponen en práctica el valor del cliente. Por otra parte, durante la Etapa 1 de un ecosistema de negocios, a menudo vale la pena cooperar.

El surgimiento de la computadora personal es un ejemplo revelador de desarrollo de negocios ecológicos. A principios de la década de 1970, una nueva tecnología - el microprocesador - surgió con el potencial de generar grandes y nuevas aplicaciones y reducir drásticamente el costo de la informática. Sin embargo, esta innovación se sentó en estado latente durante varios años. En 1975, las máquinas de aficionados como el Altair e IMSAI habían penetrado en un mercado estrecho. Sin embargo, estos equipos no eran los productos que pudieran ser utilizados por la persona promedio.

A partir de finales de 1970, Tandy Corporation, Apple, y otros introdujeron las primeras versiones de lo que eventualmente se convertiría en la computadora personal. La semilla de la innovación que todos escogieron fue el microprocesador, pero también los primeros diseñadores se deberían de crear otros productos y nuevos servicios para llevar todo el paquete junto. Estos van desde los componentes de hardware a software como a los servicios de distribución y atención al cliente.

Apple y Tandy tenía cada uno, una estrategia diferente para crear un rico ecosistema completo. Apple trabajó con socios de negocios y habló de "evangelizar" para fomentar la co-evolución. La empresa se centraba en su diseño básico de la computadora y el software del sistema operativo, esto animó a los desarrolladores independientes de software a escribir programas para sus equipos. Apple también colaboró con revistas independientes de informática, tiendas de informática, y las instituciones de formación - e incluso sembró una serie de distritos escolares con Apple IIS.

Tandy, por su parte, adoptó un enfoque más integrado verticalmente. Se trató de comprar y luego poseer su software, que van desde el sistema operativo a los lenguajes de programación y aplicaciones como los procesadores de texto. La compañía controla las ventas, servicios, soporte y formación, y el desarrollo del mercado mediante la venta exclusivamente a través de sus tiendas de Radio Shack. Por lo tanto, Tandy es más simple y su ecosistema no transmitió la emoción, las oportunidades, y las rivalidades internas de Apple, ni tampoco supo aprovechar la mayor cantidad de capital y talento a través de la participación de otras empresas.

En la Fase 2, los ecosistemas de negocios se expanden para conquistar nuevos y más amplios territorios. Los ecosistemas pueden ser rivales muy igualados y elegir atacar el mismo territorio, las batallas directas donde se comparta el mercado es muy común. La lucha se puede poner fea, ya que cada ecosistema intenta ejercer presión sobre los proveedores y clientes a unirse al grupo.

Al final, el ecosistema pueda triunfar o pueden llegar a acuerdos semiestables entre rivales para compartir el mismo ecosistema.

En general, son necesarias dos condiciones para la expansión dentro de la Fase 2:

1. Un concepto de negocio que un gran número de clientes valorarán.
2. La posibilidad de ampliar el concepto para llegar a este mercado tan amplio.

Durante la fase de expansión, las empresas establecidas pueden ejercer un enorme poder en marketing y ventas, así como en la gestión producción y distribución a gran escala, literalmente aplastar los ecosistemas más pequeños en el proceso. Uno de los retos de gestión más importantes en la etapa 2 es estimular la demanda del mercado sin sobrepasar en gran medida su capacidad para cumplir con ella.

En la Etapa 3, el poder de la negociación y ponerla en práctica es necesario para poder sobrevivir en el ecosistema. A veces, este proceso se puede establecer por contrato o por medio de la protección de patentes. Pero fundamentalmente, debe depender e la constante innovación.

Para las empresas dominantes, las fases de expansión y liderazgo de un ecosistema pueden hacer que la empresa suba o se rompa. En la Etapa 3, los productores que lideran el mercado deben extender el control de la empresa marcando el futuro de la esta y pensar en las inversiones de los principales clientes y proveedores. Y por muy buenas que sean las ganancias, cualquier empresa en el ecosistema - Líder o Seguidor - debe mantener el poder de negociación frente a otros miembros.

La Fase 4 de un ecosistema de negocios se produce cuando las comunidades de negocios maduros se ven amenazados por el aumento de nuevos ecosistemas e innovaciones. Esta fase podría simular las condiciones de la comunidad cuando sucede un terremoto: Nuevas condiciones ambientales repentinas que incluyen cambios en las regulaciones gubernamentales, patrones de compra de los clientes, o condiciones macroeconómicas. Un ambiente alterado suele ser más hospitalario con los ecosistemas de negocios nuevos o anteriormente marginales.

Si una empresa se encarga de liderar las generaciones siguientes a base de buenas innovaciones, claramente será un éxito dentro de un ecosistema y tendrá la capacidad para renovarse.

Tres enfoques generales regulan la auto-renovación, que pueden ser utilizado solo o en conjunto:

1. Las empresas dominantes pueden tratar de frenar el crecimiento de un nuevo ecosistema.
 2. Se puede tratar de incorporar innovaciones en sus propios ecosistemas
 3. Pueden reestructurarse para tratar de hacer frente a una nueva realidad.
- [8]

Las empresas deben tener en cuenta estas fases para un desarrollo óptimo dentro de los ecosistemas de negocios.

3.1 Ecosistema Biológico

De acuerdo al diccionario The New Shorter Oxford English Dictionary (1993), un ecosistema biológico es un sistema de organismos que ocupan un hábitat, junto con aquellos aspectos físicos del medio ambiente con el cual interactúan.

Como los sistemas de evolución, "los ecosistemas son dinámicos, en constantemente movimiento, en respuesta a las perturbaciones naturales y la competencia entre las especies, ya que un ecosistema tiene que estar todo el tiempo listo para responder a los cambios dentro y fuera de ella, debe haber una serie de distintas especies para garantizar que al menos parte de ellos puede hacer frente a cualquier situación nueva.

3.2 La economía como un ecosistema.

Michael Rothschild, un economista estadounidense afirma que "una economía capitalista puede ser mejor comprendido como un ecosistema vivo. Fenómenos fundamentales se observan en la naturaleza - la competencia, la especialización, la cooperación, la explotación, el aprendizaje, el crecimiento, y varios otros - son también importantes en la vida de los negocios".

Rothschild dibuja una variedad de analogías entre fenómenos económicos y biológicos. "Cada organismo es definido por la información contenida en sus genes, de misma forma cada ser vivo también se define por las relaciones con su presa, los competidores y depredadores. De la misma manera, una organización se define por su tecnología y por sus asociaciones con sus proveedores, competidores y clientes. Desde una perspectiva bionómico, organismos y organizaciones son nodos en redes de relaciones. A medida que pasa el tiempo y avanza la evolución, algunos nodos son eliminados y otros nuevos surgen, cada organismo y cada organización se mantienen en su lugar, persiguiendo el mismo objetivo.

3.3 Ecosistema Digital de negocios. (DBE)

Un Ecosistema Digital de Negocios es un entorno financiado por la Unión Europea, que proporciona una estructura, donde un software cifrado de las PYMEs europeas puedan actuar como organismos en un ecosistema. El objetivo principal es mejorar las posibilidades de que las PYME puedan competir con empresas de software más grandes.

Un ecosistema de negocios digitales se construye cuando la "adopción de tecnologías basadas en Internet para los negocios" está en un nivel donde "los servicios empresariales y los componentes de software son compatibles con un entorno de software generalizado, lo que demuestra un comportamiento evolutivo y auto-organizativo"

Desde un punto de vista ecológico dentro del ecosistema digital de negocios, podemos decir que se considera que es un "entorno digital" poblada por "especie digitales". Estas especies digitales pueden ser componentes de software, aplicaciones, servicios, conocimientos, modelos de negocio, módulos de formación, los marcos conceptuales, leyes, etc

3.4 Definición de Ecosistema de Negocios.

Como una definición concluyente consideramos un ecosistema de negocios, una estructura dinámica que consiste en una población de organizaciones interconectadas. Estas organizaciones pueden ser pequeñas empresas, grandes corporaciones, universidades, centros de investigación, organizaciones del sector público y otras partes que influyen en el sistema.

En diferentes textos, ecosistema empresarial se define ya sea de varias organizaciones o de una sola organización. En este último, la organización individuo debe funcionar como un ecosistema, con el fin de sobrevivir. Definimos ecosistema de negocios para contener una población de organismos.

Si nos atenemos a los principios de la complejidad del ecosistema de negocios deben ser auto-sostenibles. Esto significa que no se necesitarían intervenciones del gobierno para sobrevivir en los mercados locales o globales. Un ecosistema de negocios se desarrolla a través de la auto-organización, emergencia y co-evolución, que ayudan a adquirir las capacidades de adaptación. En un ecosistema de negocios se presentan simultáneamente dos elementos muy importantes, la competencia y la cooperación.

4 El ecosistema de Apple/iOS.

Apple Inc. nace en el año de 1976 de la mano de dos ingenieros, Steve Jobs y Steve Wozniak. Ese mismo año construyen su primer ordenador, el "Apple I" el cual solo logro vender pocas unidades. Un año más tarde sale a la venta el ordenador "Apple II" con el cual Jobs y Wozniack empiezan a tener éxito alrededor del mundo. A lo largo de 10 años el ordenador "Apple II" tuvo diferentes versiones y lograron vender más de 1 millón de unidades.

Poco a poco se iba creando un ecosistema dentro del mundo de los ordenadores y empezaron a salir empresas como IBM, la cual estuvo arriba de Apple, fue entonces cuando Jobs y compañía deciden sacar el proyecto Macintosh a la luz, este ordenador fue totalmente revolucionario, fue el primero en incorporar unidad de discos de 3.5 pulgadas y memoria de 400 Kb, usaba un sistema de imágenes graficas permitiendo así que los recursos del ordenador como impresoras, archivos, discos, etc., se representaran por medio de iconos gráficos, también fue el primer ordenador en tener "mouse" ó "ratón".

A mediados de los 80 la empresa tuvo varios problemas financieros y Jobs decidió irse de la empresa. Después de varios años en crisis y con problemas muy fuertes dentro de la compañía deciden traer de nuevo a Steve Jobs en el puesto de presidente y al mismo tiempo que salía la próxima generación de ordenadores, los iMac.

Luego de sus innovadores y fascinantes modelos de ordenadores personales y de escritorio, Apple decide abrir su panorama y entran al mundo de la música, con su tienda online iTunes y con un dispositivo que cambiaría el mundo de la música digital, el iPod. A partir de aquí, Apple empieza a posicionarse como una empresa líder en el sector de la informática. Pero no solo le bastó ser la empresa que generaba más dinero en la industria de la música digital, en el año 2007 Jobs anuncia un producto con el cual extendía su imperio y deciden entrar al mercado de los dispositivos de telefonía móvil con su famoso iPhone, un equipo que combina reproductor multimedia, telefonía móvil y dispositivo móvil de navegación, debido al gran éxito obtenido por el iPhone, Apple decide sacar una tienda de aplicaciones para sus dispositivos, la App Store, donde se puede bajar aplicaciones e incluso dejan que desarrolladores externos usen su plataforma y lenguajes para crear aplicaciones.

A principios del 2010 con la entrada de la era digital, de un mundo hiperconectado, Apple anuncia la llegada de su nuevo gadget, el iPad, la primer tableta de éxito comercial el cual es un dispositivo del tamaño de un cuaderno enfocado al acceso y navegación con las mismas características que los dispositivos anteriores.

Apple, al día de hoy es la empresa más grande y rica a nivel mundial. Pero esto se debe no solo a sus productos, si no, en mayor parte a como manejan su ecosistema de negocios, que a continuación comentaremos.

4.1 Modelo de negocio de Apple.

Apple es una empresa muy especial, algo muy conocido por todos los consumidores de tecnología saben que Apple siempre trata de diferenciarse de los demás, su negocio está expuesto a la luz pública. Apple gana mucho dinero enfocado en una sola acción, en la venta de sus productos, pero detrás de eso vienen una serie de aspectos a considerar que hacen que sea una empresa muy exitosa.

Algo único de Apple que lo hace muy especial y exitoso es que la empresa tiene el control absoluto de sus productos, de su fabricación y su utilización por parte de sus clientes. El sistema operativo Mac OS para los ordenadores y el iOS para el iPhone, iPod, iPad y Apple TV está diseñado por la misma empresa y de la misma forma la plataforma, el lenguaje y el software para hacer aplicaciones es de ellos. No es un producto abierto como el de otras compañías, para poder subir una aplicación al marketplace es necesario que pase los filtros de calidad de los desarrolladores de la misma empresa. Más adelante hablaremos sobre las especificaciones técnicas que maneja Apple, sus ventajas y desventajas.

Otro punto por el que Apple es exitoso es el diseño de sus productos, sencillos y con materiales que ningún otro fabricante usa, como es el aluminio en este caso.

Otro de los puntos clave del éxito de Apple se debe a los contenidos que proporciona tanto en iPod con el iTunes, en el Apple TV con videos, podcast, Series de Televisión, Películas, en el iPhone un sinnúmero de Aplicaciones de todo tipo y en el iPad una manera más fácil de poder leer libros, revistas, periódicos.

Apple se ha convertido en una “marca de culto” para la sociedad, hacen sentir al usuario como si fuera dueño de la marca, Apple no vende productos, vende un estilo de vida, también su éxito cae en la experiencia del usuario, como son el diseño de las tiendas, la relación vendedor-cliente que ofrecen dentro de las mismas, el empaquetado, todo con un diseño innovador y minimalista. Incluso, la imagen que da el CEO de la empresa es esencial para poder ser exitoso.

Las claves del éxito de Apple son:

- Nunca ser los primeros en el mercado, Apple no lanza nuevas categorías ni productos revolucionarios, entra en lo que se conoce como la fase de madurez de cualquier producto, limitándose a mejorarlos.
- Apple tiene visión a futuro, esto quiere decir llevan anticipación en muchas decisiones eso es lo que les hace tener una mayor ventaja sobre otras empresas. Como por ejemplo, la estrategia de firmar un acuerdo con Microsoft para poder implementar el Office en los ordenadores Mac, ya que todo mundo usa dicha paquetería.

- Las keynotes, estas son las presentaciones que hace Apple a lo largo del año donde presentan sus productos. La keynote más importante es la MacWorld. Es la conferencia tecnológica más esperada a lo largo del año. Las keynotes son el punto culminante del término “Hype” es la acción de crear un rumor y especular sobre un producto nuevo, a veces provocadas intencionalmente. Estas presentaciones se pueden seguir de manera online. Se presentan músicos de fama internacional al final de cada presentación. Si las keynotes cumplen con las expectativas del mercado las acciones suben de manera notable, si no, caen drásticamente.
- Da poder a los “early adopters” o los usuarios tempranos- este tipo de usuarios están ansiosos por ayudar y se identifican con la marca, hablarán muy bien de la marca y correrán el mensaje a los que los rodean.
- Apple vende estilo de vida, no productos.
- El plan estratégico de distribución, es excelente, crean sus propias tiendas como si fuera una experiencia única.

Apple ha evolucionado en cuanto a las personas a quien va dirigido sus productos, al principio los clientes eran fotógrafos, diseñadores, artistas, músicos debido al media software que incorporaban sus productos, además de que no todo mundo podía comprar ordenadores Mac, de hecho se creía que los que tenían Mac eran personas Elite, para ese entonces sus productos eran muy caros.

La llegada de los dispositivos iMac, iPod, iPhone y iPad abrió el sector de sus consumidores, los apasionados por la música, profesionistas de cualquier tipo, también al público joven con su diversidad de aplicaciones para los niños como juegos, incluso a las personas de mayor edad donde pueden leer sus libros favoritos. Cada día más personas se sienten identificadas por Apple, como lo he dicho antes, Apple vende un estilo de vida, el cual infinidad de personas quieren seguir.

Apple se diferencia de los demás al ser un producto innovador, con el que mucha gente se distingue ya que su interfaz es muy sencilla y muy amigable.

Apple consigue y conserva a sus clientes diciéndoles que son únicos, que los que comprenden sus productos no serán uno más del montón.

4.2 Marketplace

El Marketplace de Apple se llama App Store el cual es un servicio para los dispositivos iPhone, iPod, iPad, y Mac donde permite a los usuarios descargar todo tipo de aplicaciones o música y videos por medio del iTunes store. Dichas aplicaciones suelen tener un costo, pero podemos encontrar un par de ellas de manera gratuita. En un principio Apple no esperaba tener de ganancias de dicho servicio, la empresa otorga el 70% de los ingresos al vendedor la aplicación y el 30% le corresponde a Apple.

App Store nació hace 5 años, en el año 2008 debutando en 62 países con tan solo 500 aplicaciones, hoy en día existen más de 900 mil aplicaciones y la App Store se ha llegado a 155 países.

Antes de la App Store no existía nada parecido, fue la primera tienda de aplicaciones de este tipo, cambio al mundo, al usuario le encanta la experiencia de compra y las increíbles aplicaciones dentro de esta. Fue la primer tienda en masificar las aplicaciones, hacerlas negocio y la manera en la que interactúan con los dispositivos móviles. Afectó la vida del consumidor y la del desarrollador creando una industria alineada a un solo producto.

Hasta el día de hoy van 50 millones de descargas dentro de la App Store. Las primeras aplicaciones y las que cuentan con mayor número de descargas son Facebook, Google Maps, Instagram, Shazam y el juego de Angry Birds.

Poco después de la salida del primer iPhone, se desarrolló un proceso llamado Jailbreak con el cual se eliminan las limitaciones impuestas por Apple dentro del iOS mediante el uso de kernels modificados.

El Jailbreak permite a los usuarios acceder al sistema operativo permitiendo descargar aplicaciones que no estén disponibles en la App Store. Este proceso es legal en Estados Unidos pero Apple anunció que la práctica de dicho proceso daría invalidez a la garantía del producto. Han existido diferentes versiones de dicho jailbreak, cada uno diferente según el dispositivo y versión del software. El último que se ha desarrollado se llama evasi0n el cual funciona para el iOS 6 y sus actualizaciones.

Hoy en día con la App Store cualquier persona puede hacer su propia aplicación. Con un precio de menos de 100 dólares al año una persona puede desarrollar una aplicación y mandarla a Apple. El iPhone SDK era la librería para todos aquellos desarrolladores que quisieran hacer una aplicación usando el software llamado Xcode, del cual hablaremos mas tarde.

La grandeza de la App Store reside en el simple hecho que es muy fácil hacer aplicaciones.

4.3 Modelo de Desarrollo

En esta sección hablaremos sobre el enfoque técnico para el desarrollo de aplicaciones, como había comentado anteriormente, el software utilizado para el desarrollo de aplicaciones se llama Xcode.

Xcode es un conjunto de herramientas para la creación o desarrollo de aplicaciones para el Mac OS y el iOS, estas herramientas sirven para crear aplicaciones de una manera más rápida, eficiente y de alta calidad. Sirve para desarrollar aplicaciones en varios lenguajes que funcionan sobre las diversas plataformas de Apple. Puede compilar códigos de C, C++, Objective C, Java, etc.

Xcode está en contacto con los frameworks Cocoa y Cocoa Touch para crear un entorno fácil de usar y productivo. Xcode integra todas las herramientas necesarias para el desarrollador incluida la posibilidad de hacer debugging, esto quiere decir, solucionar problemas de los códigos por los mismos desarrolladores.

Las ventanas dentro de Xcode funcionan como workspaces, los cuales se pueden personalizar para que se adapte el diseño a como nosotros queramos, estas contienen un editor de texto, editor grafico, librería multimedia entre otras cosas. Otra de las ventajas de los espacios de trabajo o workspaces es la posibilidad de hacer varios trabajos a la vez.

Una de las principales ventajas de Xcode se centra en el rendimiento, mientras estas programando "Live Issues" te va avisando inmediatamente de los posibles errores dentro del código. Existe una plena interacción con la página web y los desarrolladores.

4.3.1 Compilador Apple LLVM

Este es el compilador que proporciona Apple en la herramienta Xcode. La principal característica de este compilador es que de forma constante está analizando lo que uno va escribiendo, identificando errores y proporcionando posibles soluciones, otros compiladores te avisan lo que está mal, LLVM lo corrige por ti.

4.3.2 Instruments

Instruments sirve para ayudar a localizar los cuellos de botellas dentro del rendimiento en Mac OS y en iOS.

Esta herramienta re copia en tiempo real los datos, memoria o tiempo de uso del CPU. Estos datos detectados se muestran gráficamente como pistas en el tiempo lo cual hace más fácil la localización de fallas.

4.4 Objective-C.

Al igual que C++, Objective-C es una extensión de C para hacerlo orientado a objetos, pero a diferencia de este, es que Objective-C está basado en ideas del mundo de Smalltalk, lo cual hace que Objective-C sea un lenguaje más limpio, pequeño y rápido de aprender.

Sin embargo Objective-C es un lenguaje mucho menos usado que C++. El mundo de Mac OS X es quizá una excepción a esta regla debido a que Objective-C es el lenguaje utilizado para programar en Cocoa, la nueva API orientada a objetos de Mac OS X que pretende mejorar la antigua API de programación Carbon.

Objective-C no ha sido estandarizado por ningún organismo internacional, si no que fue NeXTSTEP y ahora Mac OS X quienes han contribuido a crear este lenguaje. NeXT la empresa creó NeXTSTEP cedió la implementación de Objective-C a las GCC y ambos están ahora trabajando en mantener y mejorar el lenguaje.

Debido a que Objective-C es una extensión de C compatible hacia atrás muchas características de la sintaxis de C han sido heredadas por Objective-C entre ellas:

- Sentencias de control de flujo
- Los tipos de datos fundamentales, estructuras y punteros.
- Conversaciones implícitas y explícitas entre tipos.
- Los ámbitos de las variables- Globales, estáticas y locales.
- Las funciones y su sintaxis.
- Las directivas del preprocesador, aunque veremos que Objective-C añade más directivas del preprocesador y también añade las llamadas directivas del compilador.

Los ficheros de código de Objective-C tienen la extensión .m y en ellos podemos usar tanto la sintaxis de C como la de Objective-C. Además a partir del las GCC 2.95 podemos compilar en lo que se ha llamado Objective-C++, que no es más que la sintaxis de Objective-C mas la de C++. Los ficheros Objective - C++ se caracterizan por tener la extensión .mm o .M. La ventaja de esto es que se podrán acceder a las librerías escritas en C++ desde Objective-C y viceversa.

La principal característica que diferencia a Objective-C de otros lenguajes, es que es un lenguaje muy dinámico, en el sentido de que muchas decisiones que muchos lenguajes toman en tiempo de compilación Objective-C las deja para el tiempo de ejecución.

La gran ventaja de este dinamismo se aprecia en las herramientas de desarrollo, donde las herramientas tienen acceso a todo el runtime del programa, con lo que las herramientas de desarrollo pueden instanciar los objetos del programa, representarlos visualmente, personalizarlos, monitorizarlos y depurarlos de forma muy cómoda para el programador. [9]

4.5 Plataforma de Ejecución.

En esta sección hablaremos sobre el sistema operativo de los terminales que ejecutan las aplicaciones. El sistema operativo en donde nos enfocaremos, es el iPhone OS.

El iOS es el sistema operativo móvil de Apple, está incluido en los dispositivos iPhone, Ipod Touch, iPad y Apple TV.

Este sistema operativo nació a principios del año 2007 originalmente desarrollado para el iPhone, el primer móvil de la empresa Apple Inc. después utilizado para iPod Touch y iPad. Este sistema operativo es una variante del Mac OS X, esto significa que usa un Kernel XNU basado en Mach y utiliza frameworks Cocoa.

El iOS presenta cuatro capas de separación:

1. Núcleo del Sistema Operativo (Core OS): realiza la gestión de controladores, Memoria virtual, sistema de ficheros, TCP/IP, sockets, seguridad, gestión de memoria y comunicación entre procesos entre otras funciones.
2. Capa de Servicios (Core Services): Proporciona los servicios fundamentales del sistema con la que todas las aplicaciones van a necesitar. Realiza conexiones a la red, permite el acceso a ficheros, geolocalización, acceso a bases de datos.
3. Capa de Servicios Multimedia: El conjunto de frameworks y librerías que forman esta capa permiten construir aplicaciones con gráficos avanzados, reproducción de audio, video, etc.
4. Capa Cocoa Touch: El conjunto básico de herramientas que permiten crear y acceder a los objetos y estructuras de datos básicos, creación de interfaces de usuario, conectar la interfaz con controladores para manejar eventos, etc.

Este sistema Operativo aparece con el primer iPhone en verano del 2007, tuvo varias revisiones y actualizaciones, fueron 9 en 1 año, incluidas también en la primera generación del iPod Touch. En la versión número 2 es una versión muy corta, aparece con el iPhone 3G hasta, 6 versiones en año y medio. En la versión 3 empiezan a aparecer nuevas aplicaciones y más llamativas, durante esta versión estuvo la aparición del primero iPad. La versión 4 se estrena con el iPhone 4, esta versión es la que más aporte a hecho al sistema y a las aplicaciones. En esta versión se desarrollaron varias actualizaciones para dispositivos en específico, por ejemplo la versión 4.0.1 para el iPhone, la versión 4.0.2 para Ipod Touch y iPhone y la versión 4.1 solo para Apple TV.

La versión 5 sale con el Iphone 4S y con el Ipad 3er Generación. La versión 6, la cual, para muchos conocedores del mundo Apple ha sido la mejor, esta versión llega con el iPhone 5, está ha tenido 7 versiones desde que apareció en Septiembre del 2012. Se espera que la nueva versión, el iOS 7, llegue a finales del 2013, con una serie de nuevas mejoras, un cambio de interfaz, más amigable y el cambio radical será a nivel de imagen.

4.6 Dispositivos.

Los dispositivos que usan el iOS como dije anteriormente son el iPhone, iPod Touch, iPad y Apple TV.

El iPhone fue el primero en presentar el iOS, es el dispositivo móvil de Apple. Al día de hoy existen 6 modelos, el iPhone, iPhone 3G, 3GS, 4, 4S y el último en salir a la venta es el iPhone 5. Su memoria puede variar, hay de 8GB, 16GB, 32GB y 64GB.

Su hardware en cuanto a pantalla e interfaz es muy amigable y muy agradable a la vista, el sonido suele ser bastante bueno, la batería no es sustituible como en los demás dispositivos móviles, se carga mediante un cable USB al ordenador o directamente al enchufe, cuenta con cámara, en el iPhone 5 es de 8 Mega píxeles.

El software como hemos dicho en todos los dispositivos Apple usan el iOS, suele venir con aplicaciones como mensajes, calendario, mapas, bolsa, notas de voz, música, safari, app store entre otras.

5 El ecosistema de Android/Google.

Android Inc. es creada por el desarrollador Andy Rubin a partir de querer desarrollar un sistema operativo móvil basado en Linux con la filosofía del Open Source. Android surge como una idea nueva, un Sistema operativo móvil, adaptable a cualquier hardware y que ofreciera un entorno de desarrollo único que permitiera crear aplicaciones.

Mientras Rubin estaba consiguiendo inversionistas, se entero que otros desarrolladores estaban trabajando en un proyecto llamado Symbian, así que este se acercó a Google, les mostro el sistema operativo y Larry Page, creador de Google, compro Android Inc. y la incorporo a la sección de desarrollo de plataforma para móviles dentro del gigante Google y así Google pudo entrar en la competencia del mercado de la telefonía móvil.

Android se presentó en el 2007 dentro de la fundación que lidera Google, Open Handset Alliance, la cual es una alianza comercial de 84 compañías que se dedican a desarrollar estándares abiertos para dispositivos móviles.

El primer producto presentado con Android fue un HTC con una plataforma móvil basada en la versión del kernel Linux 2.6.

Android ha tenido varias actualizaciones a lo largo de su historia con las cuales ha mejorado el sistema operativo. La última versión del Android es el llamado Jelly Bean 4.2.

5.1 Modelo de Negocio.

El modelo de negocio de Google para ganar dinero con Android consta de varias partes involucradas, hablaremos primero sobre los desarrolladores. Android gana dinero gracias a sus aplicaciones, al igual que Apple, un desarrollador crea su aplicación y Google la vende repartiendo un porcentaje entre las dos partes involucradas. En Android se pueden hacer aplicaciones muy sencillas de manera simple. Esto a Google le encanta, porque mientras más aplicaciones existan más usuarios estarán comprando sus teléfonos ya que hoy en día las aplicaciones nos hacen la vida más sencilla.

Otra parte involucrada y muy importante son los fabricantes, ya que estos se comprometen a incluir Android es sus teléfonos a cambio de compartir las ganancias de dichos dispositivos. Google paga para que los fabricantes más poderosos del mercado traigan Android.

Los anunciantes juegan un papel muy importante dentro del modelo de negocio de Android/Google. Aquí es donde se genera el mayor porcentaje de ingresos, ya que al ser Android el sistema operativo por default en varios dispositivos de marcas poderosas, tiene mucho impacto en la sociedad tecnológica, ponen anuncios en el motor de búsquedas, inclusive en ciertas aplicaciones se puede ver publicidad y esto es muy bueno porque los desarrolladores pueden incluir publicidad en sus proyectos, así los anunciantes ganan cierta presencia dentro de las aplicaciones y Google sigue llenándose de dinero. En pocas palabras el modelo de negocio de Android/Google proviene de la publicidad móvil.

5.2 Marketplace

El Marketplace de Android/Google se llama Google Play, esta aplicación viene instalada en todos los dispositivos que tengan el sistema operativo Android. Este es un servicio que proporciona la capacidad de poder comprar o bajar de forma gratuita, aplicaciones, música, libros, juegos, series y películas.

Esta tienda de aplicaciones anteriormente llamada Android Market nace en el verano del 2008 permitiendo que solo Estados Unidos y Reino Unido pudieran desarrollar aplicaciones, a finales del 2010 Android market se encontraba en 29 países. A principios del año 2012 junta todas sus aplicaciones como Google eBookStore, GoogleMusic, GoogleMovies en una sola, para competir con iTunes de Apple creando Google Play.

Al día de hoy según un estudio realizado, Google Play ha superado en número de aplicaciones a su máximo rival, la App Store. [10]

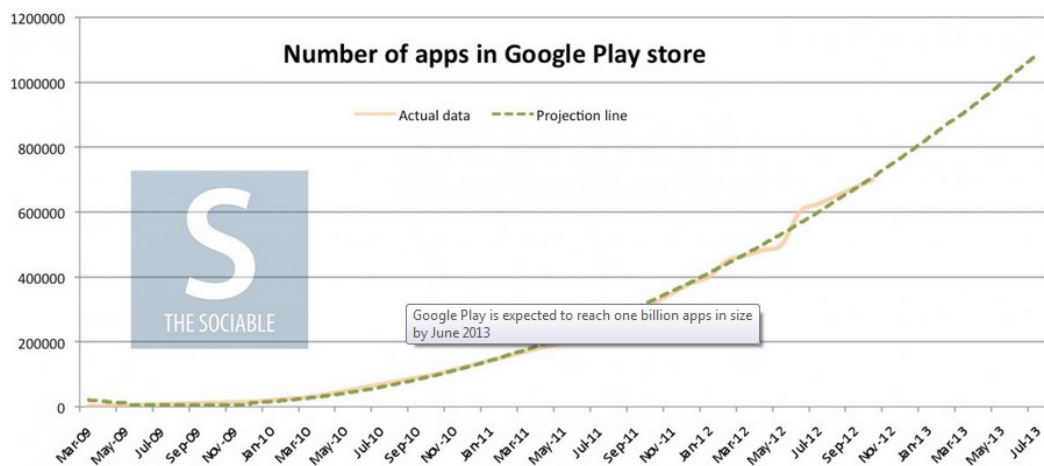


Figura 7: Número de Aplicaciones en Google Play

La tecnología que es usada en Google Play está basada en la famosa “Cloud Computing” esto significa que el usuario dispone de sus aplicaciones o datos en un servidor especialmente de Google Play donde se almacenan de manera gratuita.

5.3 Modelo de Desarrollo.

El lenguaje de programación que se usa en Android es Java. Java es un lenguaje de alto nivel, esto quiere decir que no importa la arquitectura del ordenador en el que se esté programando, por ejemplo un ordenador de 64 o 32 bits, esto permite que los programadores no se preocupen de instrucciones en lenguaje binario ya que Java trabaja como traductor.

Java es un lenguaje orientado a objetos y es multiplataforma, puede trabajar en Linux, Windows y Mac OS y en sistemas operativos móviles como es el caso de Android. Java es gratuito y se puede descargar desde su página.

Java sirve para crear aplicaciones, este lenguaje está implementado en sitios web, teléfonos móviles, en consolas de videojuegos.

Para poder desarrollar un proyecto en Android, se necesitan varios componentes, uno de estos es Java como hemos mencionado anteriormente, y explicado su funcionamiento, para programar en dicho lenguaje se necesita la interfaz JDT Plugin. Esta interfaz facilita al usuario varias herramientas para el correcto desarrollo de aplicaciones. Otro componente es el DDMS (Dalvik Debug Monitor Server) el cual es una herramienta de línea de comandos, la cual proporciona acceso directo al dispositivo si es el emulador de dispositivo virtual o dispositivo físico. Esta herramienta sirve para observar y gestionar los procesos y subprocesos que se ejecutan en el dispositivo. Otra parte importante es el Emulador, el cual proporciona una vista especial para comprobar si la aplicación se ha desarrollado de manera eficiente y sin problemas.

Al generar un nuevo proyecto Android, dado que se está utilizando un entorno Eclipse, se genera la distribución de carpetas que llevará la aplicación automáticamente, la cual será común a todos los proyectos Android.

Dichas carpetas son:

- Carpeta src. – Recoge la totalidad de Java de la aplicación.

- Carpeta res. - En ella se encuentran los recursos necesarios para generar una aplicación Android.
- Carpeta gen. - Esta carpeta guarda archivos Java que son creados de manera automática cuando se compila el proyecto.
- Carpeta assets. - Guarda el resto de archivos necesarios para el correcto funcionamiento de la aplicación, como los archivos de datos o de configuración.

5.4 Plataforma de ejecución.

Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también se usa este sistema operativo en tabletas, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma. Las versiones disponibles de Android son las siguientes:

- Cupcake: Android Versión 1.5
- Donut: Android Versión 1.6
- Eclair: Android Versión 2.0/2.1
- Froyo: Android Versión 2.2
- Ginger Bread: Android Versión 2.3
- Honey Comb: Android Versión 3.0/3.4
- Ice Cream Sandwich: Android Version 4.0
- Jelly bean: Android Versión 4.1/4.1

Android trabaja en Linux, y cada aplicación utiliza un proceso propio. Se distinguen por el ID, un identificador para que solo ella tenga acceso a sus archivos. Los dispositivos tienen un único foco, la ejecución principal, que es la aplicación que está visible en la pantalla, pero puede tener varias aplicaciones

en un segundo plano, cada una con su propia pila de tareas. La pila de tareas es la secuencia de ejecución de procesos en Android. Se componen de actividades que se van apilando según son invocadas, y solo pueden terminarse cuando las tareas que tiene encima están terminadas, o cuando el sistema las destruye porque necesita memoria, por lo que tienen que estar preparadas para terminar en cualquier momento. El sistema siempre eliminará la actividad que lleve más tiempo parada. En caso de que el sistema necesite mucha memoria, si la aplicación no está en el foco, puede ser eliminada por completo a excepción de su actividad principal.

5.5 Dispositivos

El sistema operativo Android, funciona en Smartphones, Tabletas, cámaras digitales, netbook, relojes, Laptops y otros dispositivos de varios fabricantes como Motorola, Samsung, LG, HTC, ZTE, Sony Ericsson, Huawei, Toshiba, etc. siendo el sistema operativo móvil más usado en el mundo.

El primer teléfono que salió a la venta con el sistema operativo Android fue el HTC Dream. Nexus es la línea de dispositivos Android de Google en colaboración con los gigantes Samsung, HTC y LG. Los dispositivos Nexus son los que estrenan las diferentes versiones de Android.

Las principales características de los dispositivos Nexus es que no contienen modificaciones realizadas por los proveedores de telefonía, ni de los fabricantes y que estos dispositivos se pueden liberar fácilmente para permitir un mejor desarrollo y personalización del usuario final.

6 Los ecosistemas de RIM/Blackberry y Microsoft/Windows Phone.

6.1 RIM/Blackberry.

Es una empresa canadiense, antes llamada Research In Motion (RIM) empezó desarrollando redes de datos inalámbricas para Ericsson y RAM Mobile Data, fue fundada por Mike Lazaridis. Su equipo estrella en esas épocas fue el

Inter@ctive Pager 950 el cual competía con Skytel de Motorola cerca del año 2000.

A partir de ahí RIM empezó a desarrollar dispositivos con tecnologías móviles. En el año 2002 es cuando lanza su primer producto de voz y datos, el Blackberry 5810.

El nombre de Blackberry aparece como una solución de correo electrónico. En ese mismo año RIM empezó a crecer y ver buenos ingresos en relación a los dispositivos inalámbricos Blackberry.

En año 2003 Blackberry lanza al mercado su primer móvil Blackberry a color, el modelo era el 7230.

6.1.1 Modelo de Negocio

RIM/Blackberry ha tenido muchos altibajos, en algún momento de su historia llego a ser líder mundial en telefonía móvil, tuvo ideas innovadoras, sus equipos se vendían, su filosofía gustaba pero llego un punto donde se estancaron, llegaron a tener problemas legales y no se dieron cuenta que el mercado pedía otra cosa, se quedaron muy atrás en la competencia cuando Apple saco su iPhone y cuando Google seguía potenciando el uso de Android en dispositivos. Intentaron competir con el iPad de Apple sacando su Blackberry PlayBook, le cual era un dispositivo enfocado al mundo empresarial pero la respuesta no fue nada bueno y en el año del 2011 por primera vez en 9 años sus ingresos se reducían.

Sucedido esto RIM planeo un nuevo modelo de negocio, ya que no puede competir con Apple y Android, decidió centrarse en el mercado en el que siempre ha sido bueno, este mercado es el empresarial, la mayor fortaleza de Blackberry es su teclado QWERTY el cual facilita el envío de correos electrónicos.

6.1.2 Marketplace

El marketplace de RIM/Blackberry es Blackberry World en este marketplace como en todas, se pueden comprar aplicaciones, juegos, videos, música y temas para los dispositivos de dicha compañía.

Este servicio salió a principios del año 2009, se introdujo un sistema de identificación, también se podía registrar tarjetas de crédito para compra de las aplicaciones.

Al día de hoy Blackberry World cuenta con más de 120 mil aplicaciones.

6.1.3 Modelo de Desarrollo.

El lenguaje de programación que utiliza Blackberry es Java y C++.

BlackBerry Java Development Environment es un entorno completamente integrado de desarrollo y simulación para crear aplicaciones en dispositivos BlackBerry. Gracias a BlackBerry JDE, los desarrolladores pueden crear aplicaciones con el lenguaje de programación Java ME y las API extendidas de Java para BlackBerry.

BlackBerry Java Development Environment incluye las siguientes herramientas de desarrollo:

- BlackBerry Integrated Development Environment
- BlackBerry Smartphone Simulator
- Java ME y API de BlackBerry
- Aplicaciones de ejemplo

BlackBerry IDE incluye una serie completa de herramientas de edición y depuración que se han optimizado para el desarrollo de BlackBerry Java Application. BlackBerry Smartphone Simulator ofrece un entorno tipo Windows completo y está diseñado para simular interfaces de usuario y la interacción del usuario, conexiones de red, servicios de correo electrónico y sincronización inalámbrica de datos.

El paquete de componentes de BlackBerry Java Development Environment incluye las siguientes herramientas de desarrollo para el desarrollo en IDE de terceros como NetBeans o Eclipse:

- RAPC: puede utilizar este compilador del símbolo del sistema para compilar archivos .java y .jar en archivos .cod que puede ejecutar en BlackBerry Smartphone Simulator o en un dispositivo BlackBerry.
- JavaLoader: puede utilizar esta herramienta para agregar o actualizar una aplicación en un dispositivo BlackBerry para probar y para ver información sobre los archivos .cod de la aplicación.

- BlackBerry Signature Tool: puede utilizar esta herramienta para enviar solicitudes de firma de código a BlackBerry Signing Authority Tool.
- Herramienta de verificación previa: puede utilizar esta herramienta para comprobar parcialmente sus clases antes de cargar su aplicación en un dispositivo BlackBerry.
- JDWP: puede utilizar esta herramienta para depurar aplicaciones utilizando entornos de desarrollo integrados de terceros. [11]

Recientemente se han podido hacer aplicaciones en el Web, mediante HTML5 WebWorks, donde se pueden crear aplicaciones en la web para smartphones o para la tableta Blackberry PlayBook.

Blackberry WebWorks es un proyecto libre y sus características principales son la portabilidad, integración con el núcleo de las aplicaciones Blackberry y fácil acceso, incluso cuando el usuario no tiene conexión a internet disponible. [12]

6.1.4 Plataforma de Ejecución.

El primer sistema operativo de Research In Motion, salió junto con el Pager, dando acceso al correo electrónico agenda y calculadora. La segunda versión fue introducida en los PDAs que también proporcionaba correo electrónico, navegación web, agenda y por primera vez se podían utilizar juegos.

La tercera versión se dio en el primero teléfono móvil Blackberry, este sistema operativo ya soportaba Java. El cuarto fue el Blackberry OS 4, el cual salió junto con el primer teléfono a color de la empresa, proporcionaba navegación HTML más avanzada, galería para visualizar fotos y un chat.

El auge de este sistema operativo fue en año 2008 con la llegada del Blackberry OS 5, el cual junto con la llegada de la revolución tecnológica, proporcionaba internet móvil, una aplicación de correo mejorada, integración de Gmail y geolocalización.

En el 2010 sale el Blackberry OS 6, el cual se centraba en las redes sociales, en cada teléfono móvil venía preinstalado Facebook, Twitter, el popular Blackberry Messenger, etc. Al año siguiente salió la versión OS 7 pero no introdujo muchas mejoras. En este año salió la última versión del sistema operativo, Blackberry OS 10, dentro de las mejoras destacan una interfaz fluida y amigable, un teclado inteligente, y la cámara.

Las características de Blackberry OS es que orientado al ámbito profesional y empresarial y nos podemos dar cuenta de eso debido a la gestión de correos electrónicos, el teclado y la agenda.

Desarrolladores independientes también pueden crear programas para BlackBerry pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrollador de RIM.

6.1.5 Dispositivos

El sistema operativo de Blackberry, solo corre en smartphones Blackberry y en las tabletas Playbook.

Sus modelos son: Pearl, Curve, Style, Storm, Bold y Torch. Una característica de la tableta Playbook es que también soporta Android.

6.2 Microsoft Windows Phone.

6.2.1 Modelo de Negocio de Windows Phone.

El modelo de negocio de Microsoft en cuanto a su sistema operativo móvil, es similar a la de otras empresas antes mencionadas, con pocas diferencias.

Microsoft quiere ganar de la venta de aplicaciones, 70% para el desarrollador y 30% para la empresa. También genera ingresos por publicidad, el Microsoft Advertising SDK el cual sirve para que los desarrolladores puedan introducir publicidad dentro de sus aplicaciones.

Todo mundo conoce a Microsoft ya que es la empresa líder en sistemas operativos para ordenadores, así que se propuso a entrar en la batalla de la telefonía móvil, pero está muy lejos de sus competidores. Microsoft pensó que al ser líder en OS iba a ser fácil competir contra Apple o Android, pero no fue así, sigue en deuda con sus seguidores y público en general.

6.2.2 Marketplace.

El marketplace de esta plataforma se llama Windows Phone Store. Dentro de esta tienda, se puede descargar música, videos, aplicaciones. El mercado está gestionado por Microsoft, así que si un desarrollador desea hacer una aplicación debe pasar por las estrictas regulaciones de contenido y performance. Está disponible en 54 países y a día de hoy cuenta con un poco más de 130 mil aplicaciones.

6.2.3 Modelo de desarrollo.

El lenguaje de programación de Windows Phones es La plataforma Windows Phone soporta los lenguajes de programación C# y Visual Basic .NET. Esto se debe a que ambos son entendidos por el Framework .NET. Por la parte de diseño se utiliza el lenguaje Silverlight también conocido como XAML. En el caso de las aplicaciones de tipo video juegos, utilizan XNA para generar gráficas de tipo 2D y 3D e igualmente pueden hacer uso de Silverlight al mismo tiempo.

Para programar se necesita del ambiente de desarrollo Visual Studio 2010 Service Pack 1 en cualquiera de sus versiones, de igual forma se puede descargar la versión Visual Studio Express la cual es totalmente gratuita. Antes de continuar instalando más herramientas, se debe asegurar que todo esté actualizado debidamente. Si se posee un Visual Studio que no es la versión Express, se debe de instalar el Windows Phone SDK 7.1 (se recomienda descargar el archivo formato .ISO para que tenga un respaldo del programa) para tener APIs y otras herramientas.

Seguidamente se debe revisar que todas las herramientas estén actualizadas antes de instalar el Windows Phone SDK 7.1.1. Finalmente, para expandir las herramientas del SDK, habrá que instalar el Silverlight Toolkit. [13]

6.2.4 Plataforma de ejecución.

El sistema operativo que usa Microsoft en sus dispositivos móviles, es el Windows Phone o como era llamado anteriormente, Windows Mobile.

Este OS está basado en el núcleo de Windows Embeded 6.0, sucesor de Windows Mobile 6.5 siendo incompatibles. Microsoft lanza el primer sistema operativo para dispositivos móviles, Windows Phone en el año del 2010. Este realiza múltiples cambios en la interfaz de usuario como en usabilidad del sistema.

Otra característica de este OS, son los requisitos mínimos de hardware. Además este OS integra la nueva interfaz de usuario con el diseño de cuadros llamado metro, integración de servicios de la nube con el skydrive y redes sociales. En Windows Phone 7.5 incluye Internet Explorer 9, soporte para CSS3 Media queries y GPS.

Las desventajas de este sistema operativo es que no permite el cambio de país de residencia, volumen apartado para el reproductor y llamada, administrador de archivos.

Windows Phone 8 es el ultimo Sistema operativo, el cual solo es para los nuevos dispositivos, no es compatible con las versiones pasadas. Este consta de nuevo interfaz, dota al sistema de multitareas para aplicaciones de terceros, soporte de procesadores de varios núcleos, transferencia por Bluetooth, aplicaciones de código nativo, entre otras.

6.2.5 Dispositivos

Windows Phone se puede encontrar en varios dispositivos preinstalado. Los requisitos que pide Microsoft a los fabricantes para poder insertar Windows Phone al dispositivo es que tengan una cámara de 5 mega píxeles como mínimo y una resolución en la pantalla de 480x800 pixeles.

Entre los fabricantes que usan por default el sistema operativo de Microsoft son: HTC, LG, Samsung, Huawei y Nokia gracias a una asociación que se dio un par de años atrás en la que acordaron dichas empresas (Microsoft y Nokia) que Windows Phone será el principal sistema operativo de los Smartphones de Nokia.

7 El ecosistema de Firefox OS.

Un nuevo capítulo se añadió a esta historia con el debut de Firefox. Firefox es un descendiente lejano de Mosaic y Netscape, los cuales fueron navegadores. Mosaic, fue el primer navegador gráfico disponible para visualizar páginas web que después se abandonó el desarrollo de este navegador para después pasar a desarrollar Netscape por los mismos desarrolladores de Mosaic. En 1998, Netscape creó el Proyecto Mozilla e hizo su código libremente disponible del navegador como una estrategia experimental para obtener una ventaja competitiva frente a Microsoft. Esto permitió a los programadores todo el mundo para estudiar el código de Mozilla y seguir su desarrollo. Con el paso del tiempo, esta comunidad de desarrolladores siguió contribuyendo a su desarrollo. Netscape y AOL (que había adquirido Netscape) se mantuvieron fuertemente involucrados con Mozilla y lanzaron varios Netscape y AOL productos del código de Mozilla en evolución. Sin embargo, el interés en el desarrollo navegador AOL disminuyó a lo largo del tiempo ya principios de 2003 AOL decidió reducir la participación, estableciendo el proyecto libre de trazar su propio curso.

En 2003, la Fundación Mozilla se creó como una organización independiente. El escenario estaba listo para la comunidad Mozilla crecimiento de la inversión más allá de apalancamiento de Netscape en el código del navegador y hacer su propia marca. Esta apasionada comunidad se centró en la creación de un navegador de última generación llamado metas Firefox. The para Firefox eran simples: Hacer un navegador ligero que fue rápido y fácil de usar. Pondría a los usuarios de nuevo en control de su experiencia de navegación web. No sería renunciar a cualquier parte de la experiencia del usuario con molestias. También añadiría un sistema de extensión que daría a los usuarios la capacidad de personalizar, experimentar, y ajustar el navegador de mil

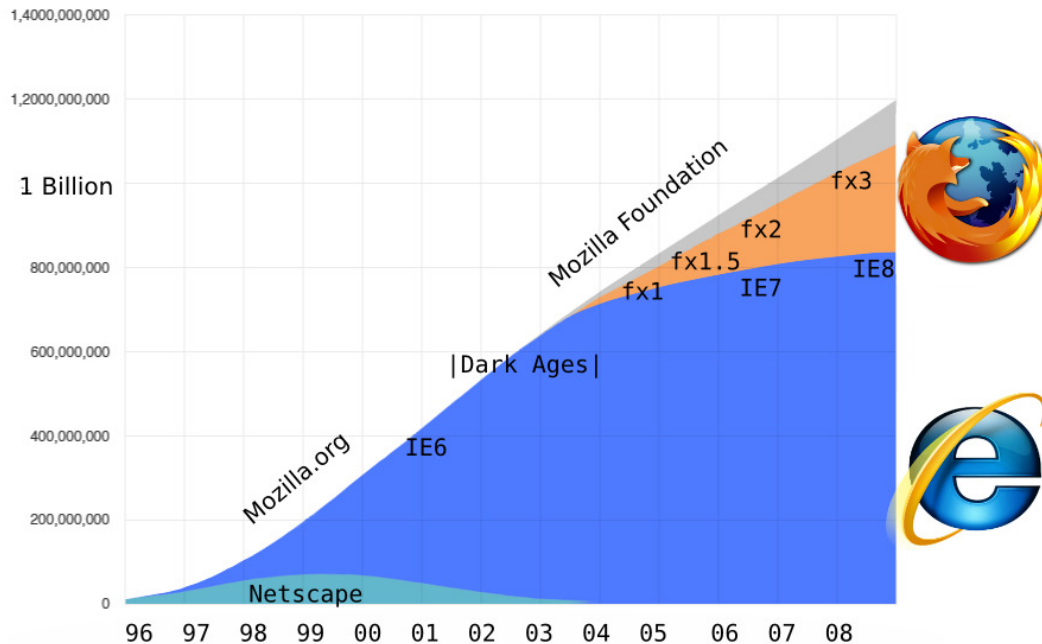


Figura 8: Comparación de descargas entre Firefox e Internet Explorer.

En 2004, Firefox 1.0 fue lanzado. Desde entonces, el número de usuarios ha crecido de forma constante. A partir de 2009, Firefox dispone de una cuota de mercado mundial del 23%, con más de 300 millones de los 1,2 millones de usuarios de Internet de todo el mundo que utilizan Firefox como su navegador web. El día de Firefox 3 fue lanzado en un evento conocido como "Download Day", más de 8 millones de usuarios descargaron la nueva versión en un solo día.

Firefox continúa innovando. Con características tales como un método sencillo para la suscripción de títulos de las noticias actualizadas de forma automática, fichas de página en el hogar que le ayudarán a obtener el contenido en la web más rápida, una función de bloqueo de pop-up, y el creciente número de extensiones que le permiten adaptar su navegador, Firefox te ayuda a mantener el control de la forma de interactuar con la web. Firefox ha despertado un renovado interés en la mejora de los navegadores web.

En respuesta a Firefox, Microsoft cambió sus planes, reinicie su desarrollo navegador y lanzó IE7 y IE8 recientemente. Apple y Google también se han involucrado en los esfuerzos para construir nuevos navegadores (Safari y Chrome, respectivamente). La innovación y la elección están regresando a los navegadores y la Web.

Firefox se ejecuta en cualquier sistema operativo y se localiza en más de 75 idiomas diferentes. Está construido por una comunidad de desarrolladores de todo el mundo que sienten pasión por mejorar el navegador y la web. Lo mejor de todo, Firefox es gratuito! [14]

Firefox OS

Firefox OS (también conocido por su nombre en clave "Boot to Gecko" o "B2G") es el sistema operativo móvil de código abierto basado en un kernel de Linux y la tecnología de Mozilla Gecko de Mozilla, esto permite a los usuarios ejecutar aplicaciones desarrolladas por completo en HTML5, JavaScript, y otras Web APIs.

Firefox OS es un sistema operativo móvil que está libre de la tecnología propietaria y al mismo es una plataforma de gran alcance que proporciona a los desarrolladores de aplicaciones una oportunidad de crear productos excelentes. Además, es lo suficientemente flexible y capaz de hacer feliz al usuario final.

Para los desarrolladores web, la parte más importante a entender es que la interfaz de usuario completa es una aplicación Web, que es capaz de mostrar y lanzar otras aplicaciones Web. Cualquier modificación que se realice en la interfaz de usuario y cualquier aplicación que desarrolles para correr en Firefox OS son páginas web, aunque con un mayor acceso a los equipos y servicios del dispositivo móvil. [15]

7.1 Modelo de negocio.

El modelo de negocio de Firefox OS es simple, ya que es un sistema operativo muy nuevo, no puede entrar en la competencia contra iOS Y Android, lo que harán es tratar de llenar los espacios vacíos que dejaron estos dos sistemas operativos.

Primero, Firefox OS a diferencia de los demás incluido Android/Google, es que es totalmente abierto y cualquiera puede realizar extensiones y mejoras. Otra estrategia es la de llegar a un mercado más joven, a los usuarios que apenas están en condiciones de tener un Smartphone, para que empiecen a familiarizarse con la tecnología.

De mismo modo este sistema operativo llegará a las personas que no cuentan con los recursos económicos para comprar un dispositivo iOS o Android y es por eso que el primer dispositivo con Firefox OS, fabricado por ZTE y comercializado por Telefónica España, salió a la venta el mes de Junio pasado a un costo de 69€ con 30€ de saldo para clientes prepago, considerando los precios de los Smartphones de iOS y Android es muy económico y eso es una buena estrategia para que todo el mundo empiece a conocerlo y vaya ganando popularidad.

Otra ventaja que tendrá Firefox OS, es que alrededor del mundo tiene varios seguidores, al igual que en los SO anteriormente mencionados existen leales seguidores del navegador Firefox, así que esas personas también comprarán el dispositivo.

Otra parte importante dentro del modelo de negocio de Firefox OS es que cuenta con el apoyo de los mejores fabricantes de dispositivos móviles, estos han accedido a crear ciertos smartphones con Firefox OS.

Lo que hace que Firefox OS sea diferente a sus competidores es que está diseñado para funcionar muy rápido en los teléfonos inteligentes de gama baja. Esto se debe a que, en lugar de cargar un sistema operativo para teléfonos inteligentes completo, como el resto de las plataformas, los smartphones con Firefox OS ejecutarán un sistema operativo de bajos recursos donde sólo soportará un navegador web y todo lo que se ejecuta será a través del navegador.

En efecto, las aplicaciones para Firefox OS son como los sitios web en que están enteramente codificadas en HTML5, aunque a diferencia de sitios web para móviles ordinarios, se puede tener acceso a hardware del teléfono como por ejemplo, activar la cámara a través de una aplicación donde no se pueda realizar esta acción dentro de una página web.

Mejor aún, a medida que se ha adoptado HTML5 en todo el sector de la industria móvil, una aplicación desarrollada para Firefox OS debería, en teoría, correr en cualquiera de las otras plataformas móviles.

Mozilla también ha añadido la capacidad para cobrar a los usuarios por las aplicaciones de los desarrolladores, o realizar compras en la aplicación a través de su tienda de aplicaciones Mozilla Marketplace. Esto significa que ha llegado

al punto en que ahora es potencialmente una plataforma de aplicación económicamente viable, en lugar de sólo un juguete para los desarrolladores.

Esta nueva plataforma podría ser una oportunidad fantástica para los emergentes móviles de nueva creación.

7.2 Marketplace

El marketplace de Mozilla es Firefox Marketplace, una tienda para aplicaciones en HTML5 donde al conectarse a la web podemos encontrar, probar y descargar aplicaciones, al día de hoy ya existen una gran cantidad de aplicaciones incluyendo las más exitosas como facebook y twitter.

Es una tienda en línea, abierta y sin derechos de propiedad para aplicaciones web construidas en HTML5 el cual permite a los desarrolladores crear aplicaciones multi-plataforma utilizando tecnología, lenguajes y herramientas web.

El Firefox Marketplace brinda aplicaciones web que se comportan igual que las aplicaciones nativas y pueden utilizarse en múltiples plataformas y dispositivos. El diseño del Marketplace también te facilita encontrar tus aplicaciones favoritas y descubrir las nuevas. [16]

7.3 Modelo de Desarrollo.

Las aplicaciones en Firefox OS son aplicaciones web escritas sólo con HTML, CSS y JavaScript. Se publican en la Web como se publicarían en cualquier otro sitio Web. Para que el sitio Web pueda instalar la aplicación en el dispositivo, sólo se tiene que añadir un manifiesto y conectar un botón de instalación.

Las aplicaciones web son aplicaciones construidas usando tecnologías web estándar. Trabajan en cualquier navegador web moderno, y se pueden desarrollar utilizando sus herramientas favoritas. Algunas de las características que distinguen a las aplicaciones web de las páginas web son: Las aplicaciones se instalan por un usuario, que son autónomos y no siempre requieren el cromo de la ventana del navegador, y pueden ser construidos para funcionar sin conexión. Gmail, Twitter y Etherpad son las aplicaciones web.

El proyecto Mozilla Aplicaciones Web Abierta propone algunas pequeñas adiciones a los sitios existentes para convertirlos en aplicaciones que se ejecutan en un rico, divertido y potente entorno informático. Estas aplicaciones se ejecutan en navegadores de escritorio y dispositivos móviles, y es más fácil para un usuario descubrir y poner en marcha de sitios Web. Ellos tienen acceso a un conjunto cada vez mayor de características novedosas, como la sincronización a través de todos los dispositivos de un usuario. [17]

7.3.1 Estructura de las aplicaciones.

“Packaged and Hosted Apps.”

Hay dos tipos de aplicaciones en Firefox OS: “Packaged and Hosted”. “Packed Apps” son esencialmente un archivo zip que contiene todos los activos de las aplicaciones como por ejemplo: HTML, CSS, JavaScript, imágenes, manifiestos, etc. Las “Hosted apps” se ejecutan desde un servidor en un dominio dado, al igual que una página web estándar. Ambos tipos de aplicaciones requieren un manifiesto válido. Cuando llega el momento de mostrar la aplicación en el Firefox Marketplace, que o bien cargar tendrás estas dos opciones, ya sea subirla en archivo zip ó proporcionar el URL de donde se carga tu aplicación.

“App Manifests”

Cada aplicación de Firefox requiere un archivo manifest.webapp en la raíz de la aplicación. El archivo manifest.webapp proporciona información importante acerca de la aplicación, como la versión, el nombre, la descripción, icono de ubicación, las cadenas de configuración regional, los dominios de la aplicación desde donde se puede instalar, y mucho más (se requiere únicamente el nombre y la descripción). Un manifiesto simple lo podemos observar dentro de la siguiente figura:

```

{
  "version": "0.1",
  "name": "Your App",
  "description": "Your new awesome Open Web App",
  "launch_path": "/index.html",
  "icons": {
    "16": "/img/icons/mortar-16.png",
    "48": "/img/icons/mortar-48.png",
    "128": "/img/icons/mortar-128.png"
  },
  "developer": {
    "name": "Your Name",
    "url": "http://yourawesomeapp.com"
  },
  "installs_allowed_from": ["*"],
  "locales": {
    "es": {
      "description": "Su nueva aplicacion impresionante Open Web",
      "developer": {
        "url": "http://yourawesomeapp.com"
      }
    },
    "it": {
      "description": "Il vostro nuovo fantastico Open Web App",
      "developer": {
        "url": "http://yourawesomeapp.com"
      }
    }
  },
  "default_locale": "en",
  "permissions": {
    "systemXHR": {}
  }
}

```

Figura 9: "Firefox Manifest"

"App Layout and Design"

El diseño se ha convertido cada vez más importante a medida que más resoluciones de pantalla se convierten en norma en diferentes dispositivos. Incluso si su aplicación se centra sólo en Firefox OS, es importante recordar que Firefox OS se puede instalar en una variedad de dispositivos que utilizan diferentes resoluciones de pantalla.

```

/* The following are examples of different CSS media queries */

/* Basic desktop/screen width sniff */
@media only screen and (min-width : 1224px) {
  /* styles */
}

/* Traditional iPhone width */
@media
  only screen and (-webkit-min-device-pixel-ratio : 1.5),
  only screen and (min-device-pixel-ratio : 1.5) {
  /* styles */
}

/* Device settings at different orientations */
@media screen and (orientation:portrait) {
  /* styles */
}
@media screen and (orientation:landscape) {
  /* styles */
}

```

Figura 10: Ejemplos de diseños para varios dispositivos.

Hay muchos marcos disponibles para JavaScript y CSS para ayudar en el diseño y desarrollo de aplicaciones móviles (bootstrap, etc), elegir los marcos que mejor se adapten a la aplicación y desarrollo del estilo.

“Tools and Testing”

Las pruebas son muy importantes en el apoyo a los dispositivos móviles. Hay muchas opciones para probar aplicaciones Firefox OS:

- *Simulador Firefox OS*

La instalación y uso del simulador de Firefox OS es la forma más fácil de empezar a trabajar con su aplicación. Después de instalado el simulador es accesible a través de las Herramientas -> Web Developer -> menú Firefox OS simulador. El simulador se lanza con una consola de JavaScript para que pueda depurar la aplicación desde el simulador.

- *Pruebas unitarias.*

Las pruebas unitarias son extremadamente valiosas cuando se prueban en diferentes dispositivos. JQuery QUnit es una utilidad de prueba del lado del cliente, y se puede utilizar cualquier conjunto de herramientas de prueba que le convengan.

- *Instalar Firefox OS en un dispositivo.*

Dado que Firefox OS es una plataforma de código abierto, el código y las herramientas están disponibles para construir e instalar Firefox OS en tu propio dispositivo.

- *Distribución de una aplicación.*

Una vez que la aplicación está completa, puede presentarse dentro del Firefox Marketplace. La aplicación será validado y se deberá elegir en qué dispositivos se apoyará su aplicación (por ejemplo, Firefox OS, Escritorio Firefox, Firefox Mobile Firefox Tablet). Una vez validado puede agregar más detalles sobre su aplicación (imágenes, descripciones, precios, etc) y se presentará oficialmente la aplicación para su inclusión en el mercado. Una vez aprobada la aplicación se encontrara disponible en el mundo para la compra y la instalación. [15]

7.4 Plataforma de ejecución.

Boot to Gecko es un proyecto que tiene como objetivo construir un sistema operativo independiente para la web.

Las principales características de Boot to Gecko son:

- Extender las capacidades de las tecnologías web abiertas.
- Estandarizar los cambios exitosos.
- Desplazar pilas de aplicaciones de proveedores propios, individuales.
- Un trabajo totalmente abierto, fuente de liberación en tiempo real.

7.4.1 Arquitectura de Boot to Gecko

Esta se divide en 3 capas:



Figura 11: Arquitectura B2G.

- Gonk- El nivel más bajo del sistema operativo B2G. Gonk consiste en un núcleo de Linux y el espacio de usuario de "Hardware Abstraction Layer" (HAL). El núcleo y varias bibliotecas de espacio de usuario son proyectos de código abierto comunes: Linux, libusb, bluez, etc. Algunas otras partes del HAL se comparten con el proyecto Android: GPS, cámara, entre otros. Se podría decir que Gonk es una simple distribución de Linux. Gonk es un "porting target" de Gecko, hay un puerto de Gecko a Gonk, al igual que hay un puerto de Gecko a OS X, y un puerto de Gecko para Android. Dado que el proyecto B2G tiene el control total sobre Gonk, podemos exponer interfaces a Gecko que no son posibles para exponer en otros sistemas operativos. Por ejemplo, Gecko tiene acceso directo a la pila de telefonía completa y el display framebuffer en Gonk, pero no tiene este acceso en cualquier otro sistema operativo.

- Gecko- El "tiempo de ejecución de la aplicación" de b2g. A un alto nivel, Gecko implementa los estándares abiertos para HTML, CSS y JS y hace que las interfaces funcionan bien en todos los sistemas operativos que soporta Gecko. Esto significa que Gecko consiste en, entre otras cosas, una pila de red, pila de gráficos, motor de la disposición, la máquina virtual (por JS), y las capas de portabilidad.
- Gaia- La interfaz de usuario de B2G. Gaia implementa una pantalla de bloqueo, la pantalla de inicio, marcador de teléfono, la aplicación de mensajería de texto, cámara de aplicación, ... y muchos más. Gaia está escrito enteramente en HTML, CSS y JavaScript. Su única interfaz para el sistema operativo subyacente es a través de las "Open Web APIs", que se implementan por Gecko. Gaia funciona bien cuando se ejecuta en la parte superior de B2G, sin embargo, ya que sólo utiliza las API de web estándar, de igual forma funciona en otros sistemas operativos y en otros navegadores web (aunque con funcionalidad degradada). Las aplicaciones de terceros se pueden instalar junto con Gaia. [18]

Gecko es el motor del navegador de código abierto diseñado para soportar los estándares abiertos de Internet, como HTML 4, CSS 1 y 2, el W3C DOM, XML, JavaScript y otros.

Gecko es usado en varios navegadores, incluyendo Mozilla Firefox, SeaMonkey, Camino, entre otros. Gecko está continuamente en desarrollo en mozilla.org. Gecko se ha conocido previamente por los nombres en clave "Raptor" y "NGLayout", el nuevo nombre fue elegido tras una disputa de infracción de la marca.

Básicamente, un motor de diseño toma el contenido (por ejemplo, HTML, XML, archivos de imágenes, applets, etc) y la información de formato (como Cascading Style Sheets, etiquetas HTML de presentación, etc) y muestra el contenido con formato en la pantalla.

Formalmente, un motor de diseño define la política de la colocación de un documento y coloca el contenido en una página. El núcleo de Gecko es un

motor renderizado muy rápido. Gecko también ofrece la posibilidad de analizar los distintos tipos de documentos (HTML, XML, SVG, etc), las capacidades avanzadas de renderizado, incluyendo la composición, transformaciones, y soporte para JavaScript y plugins.

Gecko es tan rápido y tan fuerte que se está utilizando para crear la interfaz de usuario de algunas aplicaciones ("chrome"). En otras palabras, Gecko no sólo estará mostrando el contenido del documento, sino que también puede pintar las barras de desplazamiento, barras de herramientas y los menús en la pantalla.

Gecko proporciona la base necesaria para mostrar el contenido en la pantalla, incluyendo un motor de diseño y un conjunto complementario de componentes del navegador. Sin embargo, Gecko no empaqueta todos estos componentes junto con otros módulos de interfaz de una aplicación coherente y fácil de usar (incluyendo menús, barras de herramientas, etc.), como Firefox.

mozilla.org ensambla los componentes necesarios en sus aplicaciones, como Firefox, Thunderbird, SeaMonkey, Camino, que están disponibles para su descarga gratuita desde mozilla.org. Netscape lanzó su propia versión del navegador calificado como Netscape Navigator. Otras compañías lanzan su propio software y hardware que utilizan Gecko.

Los terceros, como los ISVs y proveedores de hardware eligen los componentes que desean utilizar en sus aplicaciones o dispositivos de hardware. Ciertos componentes del navegador no se proporcionan como parte de Gecko, como los marcadores, historial, libreta de direcciones, etc.

El navegador Mozilla, lanzado por primera vez como Navigator 1.0, se desarrolló rápidamente por un pequeño equipo apasionado de la creación de la próxima killer app, y lo lograron. Ahora que la web ha evolucionado, se necesitaba un motor de diseño de nueva generación en el que los futuros productos podrían ser construidos. Gecko permite una nueva clase pionera de contenido dinámico que es más interactiva y ofrece un mayor control de la presentación de los desarrolladores web, usando estándares abiertos de Internet en vez de APIs privadas.

Gecko se encuentra en el corazón de Mozilla y Firefox, así como otros, estos alimentan todos los componentes individuales. Las tecnologías Gecko también potencian la visualización del portal mozilla.com, entrega rápidamente los contenidos y servicios. La arquitectura de Gecko le servirá a Mozilla en un futuro no muy lejano, ya que permitirá una rápida comercialización, más innovación, desarrollo menos costoso, más fácil distribución y actualización, y un mejor soporte de plataforma. [18]

7.5 Dispositivos

En este mes, Julio 2013 salió a la venta el primer dispositivo con Firefox OS, fue presentado por Telefónica España y el fabricante es ZTE. El nombre de este dispositivo es ZTE Open.

El target de este dispositivo es para las personas que apenas compraran sus primer Smartphone ya que es de bajo coste, en España costará 69€ y tendrá 30€ de saldo.

La principal característica del móvil con Firefox OS es que, aparte de algunas aplicaciones nativas (como la lista de contactos), el teléfono está diseñado con HTML, el estándar libre y abierto de la web, y tecnologías de código abierto. Concretamente, en HTML5, la quinta versión del lenguaje de la web que se ha consolidado en todo tipo de sistemas y dispositivos.

El ZTE Open cuenta con una pantalla de 3,5 pulgadas, pantalla táctil HVGA TFT y cuenta con 256 MB de RAM y 512 MB de ROM, además de cámara de 3,2 mega píxeles. Incluye además tarjeta microSD de 4 GB. En cuanto a software, ofrece todas las posibilidades de un Smartphone, como son la mensajería, el email, calendario, radio FM, cámara con filtros integrados, servicios de localización gracias a los mapas de Nokia HERE y posibilidad de navegar por Internet a través de Firefox.

8 Análisis comparativo de Firefox OS con otros ecosistemas.

La principal ventaja de Firefox sobre los demás es que es un sistema operativo basado en navegador web y esto hace que sea un sistema operativo muy ligero y consuma pocos recursos. Todo el sistema operativo está basado en HTML5 y como hemos dicho anteriormente esto nos permite ejecutar todas aplicaciones que ya estén desarrolladas con este estándar abierto. Al ser un sistema abierto se puede modificar por completo por los usuarios y desarrolladores, un poco parecido a Android.

En cuanto al modelo de negocio de Firefox con los de iOS y Android podemos decir que en estos momentos no puede entrar en la batalla que libran estos dos, Apple y Google/Android están muy por encima de Firefox, como lo comentamos anteriormente, Firefox entra para cubrir los espacios que sus adversarios han estado dejando, Firefox tratara de entrar poco a poco en la vida de las personas, primero entrando por los más jóvenes, seguido por los seguidores de Mozilla y así hasta ir llegando al público en general ya que un principio se desarrollara en una gama de teléfonos mas económica, lo cual es una estrategia muy buena para que se dé a conocer.

La principal comparación entre Firefox y los demás ecosistemas móviles es la forma en la que se desarrollan las aplicaciones dentro de los sistemas operativos.

Existen tres tipos de aplicaciones a desarrollar, uno es el desarrollo móvil web, otro es el desarrollo móvil nativo y el tercero son las aplicaciones híbridas.

Las aplicaciones móvil web son optimizadas para ser visualizadas desde un dispositivo móvil o tableta. Por definición, estas aplicaciones serán accedidas utilizando el navegador que viene por defecto dentro de los dispositivos: ya sea Safari (o algún otro) en iPhone o iPad y digamos Dolphin (entre otros) para Android.

Las principales características de las aplicaciones móvil web son:

1. Serán accedidas desde un navegador desde el teléfono.
2. Para todos los dispositivos se desplegará *casi* que de la misma forma.
3. Es prácticamente lo mismo que desarrollar cualquier aplicación o sitio web: utilizas HTML, CSS y JavaScript.

Por otra parte, el desarrollo móvil nativo es el desarrollo de aplicaciones que serán instaladas en el sistema de archivos de cada dispositivo y serán distribuidas por los mercados de aplicaciones móviles, como el AppStore de iOS o el Play Market de Android.

Las principales características de las aplicaciones nativas son:

1. Serán descargadas desde el mercado de aplicaciones para el sistema operativo: Play Market o el AppStore.
2. Es una aplicación distinta por sistema operativo: una para Android y otra para iOS (y en muchos casos una para iPhone y otra para iPad).
3. Se desarrollan con lenguajes como Java (Android) u Objective-C (iOS), pero hay alternativas como Appcelerator que permiten desarrollarlas utilizando JavaScript puro, para ambas plataformas.

Apple y Android se centran en desarrollar aplicaciones nativas.

La idea de las aplicaciones híbridas es sencilla, utilizar el desarrollo nativo cuando es mejor o es necesario (para utilizar la cámara, por ejemplo), pero utilizar tecnologías web y el desarrollo web cuando es más práctico (por ejemplo en interfaces más complejas).

Un maestro haciendo esto es LinkedIn, quien en la última versión de su aplicación móvil decidió hacerla híbrida para iOS, pero nativa en Android:

- Todas las listas infinitas en la aplicación son nativas.
- Todas las ventanas son nativas, estas funcionan como contenedores de vistas web.
- Todas las páginas que son detalles con información son JavaScript y HTML, por su fácil desarrollo.

Cuando las vistas son vistas web, se utiliza una ventana (un Web View) para mostrar el contenido o la página web. Así, nativo sólo es el contenedor o wrapper de la aplicación.

Otro caso interesante de aplicaciones híbridas es Instagram. En Instagram utilizan nativo para tomar y publicar la fotografía, pero web para desplegar las fotografías y tu perfil. Esto permite que la aplicación pueda ser accedida fácilmente sin conexión a internet para editar y tomar una fotografía, pero hace fácil a los desarrolladores mejorar la lista de fotografías sin sacar una nueva versión ya que tendrían solamente que trabajar en su servidor.

Algunas de las herramientas más utilizadas hoy en día para crear aplicaciones híbridas son:

- Phonegap: Quien permite realizar aplicaciones con HTML5, CSS3 y JavaScript, pero que serán empaquetadas como aplicaciones nativas. Seguramente si vas a utilizar Phonegap necesitarás ojear Sencha Touch o jQuery Mobile para la interfaz.
- Trigger.io: Básicamente, casi igual que Phonegap. Ellos se adjudican ser 5 veces más rápidos que Phonegap.
- Titanium Appcelerator: Utilizando Web Views para embeber un navegador web dentro de alguna ventana nativa.
- O hasta Java (+ Android SDK) u Objective-C (+ Xcode), utilizando vistas web embebidas en la aplicación.

8.1.1 Tabla comparativa entre los ecosistemas móviles.

	Modelo de Negocio	Marketplace	Modelo de Desarrollo	Plataforma de Ejecución	Dispositivos
APPLE	<ul style="list-style-type: none"> ❖ Estilo de vida. ❖ Diferenciarse de los demás. ❖ Control absoluto de sus productos. 	App Store	<p>Xcode</p> <p>Objective C</p>	iOS	iPhone, iPod, iPad, Apple TV.
ANDROID	<ul style="list-style-type: none"> ❖ Desarrollo de aplicaciones. ❖ Fabricantes. ❖ Anunciantes. 	Google Play	Java	Android	Motorola, Samsung, HTC, LG, Huawei, Sony Ericsson
RIM	<ul style="list-style-type: none"> ❖ Centrarse en el mercado empresarial. 	Blackberry World	<p>Java</p> <p>C++</p>	Blackberry OS	Blackberry y Tablet as Playbook.
MICROSOFT	<ul style="list-style-type: none"> ❖ Renombre mundial. ❖ Desarrollo de aplicaciones. ❖ Fabricantes. ❖ Anunciantes. 	Windows Phone Store	<p>C#</p> <p>Visual Basic .NET</p>	Windows Phone	HTC, LG, Samsung, Huawei y Nokia (Principal OS)
FIREFOX OS	<ul style="list-style-type: none"> ❖ Llenar espacios libres. ❖ mercado nuevo y más joven. ❖ lealtad de sus fans. 	Firefox Marketplace	<p>HTML</p> <p>CSS</p> <p>JavaScript.</p>	Boot to Gecko	ZTE Open

9 Propuestas de mejora y Trabajo a Futuro.

Ya que es un sistema operativo nuevo, un proyecto que tiene un año y que apenas el pasado mes de Julio salió a la venta al público con el ZTE Open se irán encontrando ciertas cuestiones a arreglar, deberemos esperar que tanto soporte le proporcionara los desarrolladores, los fabricantes y de igual forma los operadores.

Es totalmente nuevo, el marketplace tiene pocas aplicaciones, conforme se vaya ganando la confianza de la gente y de los desarrolladores podremos ver más y mejores aplicaciones, otra cuestión es ver cómo funciona el lenguaje HTML5 en dispositivos móviles, ya que suele tener ciertos problemas a la hora de que el móvil este en modo "offline".

Aun así, esta primera versión de Firefox OS será una catapulta para que los desarrolladores empiecen a familiarizarse con el sistema, hagan pruebas de sus aplicaciones y así desarrollar cualquier cantidad de aplicaciones web.

Como todas las primera versiones de todas las grandes marcas empezaron por una interfaz muy abstracta, muy básica y Firefox no es la excepción, en futuras mejoras deberán mejorar en la calidad de la interfaz, el diseño y además tener en cuenta a que mercado trataran de llegar después, tal vez es un par de años sean capaces de competir contra Apple y Android.

10 Conclusiones

Hoy en día las comunicaciones móviles son muy importantes para relacionarnos con la gente, entramos en una nueva etapa de la tecnología, las comunicaciones móviles y a lo largo de este documento nos podemos dar cuenta de cuánto han evolucionado, antes este crecimiento tardaba mucho tiempo pero ahora avanzan muy rápido, no podemos darnos una idea de cómo acabará.

Apple seguirá haciéndose fuerte, tiene un modelo de negocio con el cual sus productos se venden por si solos, tiene muchos seguidores y cada día crecen, su manera de desarrollar aplicaciones es única y sus dispositivos son excelentes, la cuestión es que iOS cuenta con muchas restricciones, como por ejemplo sobre el hardware y software en el que corre el SO, no permite mucha personalización en la interfaz y esto hace que sea un sistema operativo muy cerrado.

Android seguirá acaparando los sistemas operativos de la mayoría de los fabricantes, el punto fuerte de Android es que es un sistema abierto de programación y la modificación, mejora y solución de ciertos problemas será más rápido. Hoy en Google Play cuenta con más aplicaciones que Apple y esto subirá más con el paso del tiempo. El ecosistema de Android está en constante movimiento y una gran desventaja de ser libre y abierto es que las actualizaciones dependen del desarrollador y no de Android, caso contrario a lo que pasa en Apple. Del mismo modo, suele tener más riesgos debido a su poca seguridad ya que Android es de todo el mundo.

Windows Phone está donde está solo por ser el sistema operativo que se usa en casi todos los ordenadores del planeta y esto lo ha llevado a luchar por el tercer lugar dentro de esta gran batalla. Su nuevo sistema operativo el Windows Phone 8 podría tener más seguidores que sus predecesores, ya que este es más sencillo de utilizar y tiene una interfaz muy amigable, la cámara es mucho mejor ahora y tiene su propio servidor en la nube y quizá una de las desventajas que yo veo de este sistema operativo es que requiere de dispositivos móviles muy potentes y al ser relativamente nuevo dispone de pocas aplicaciones en sus marketplace, pero probablemente llegue a ser muy popular dentro de los meses siguientes.

Blackberry no está corriendo en la batalla con Android y Apple, RIM la perdió hace mucho, pero tiene a sus seguidores y está enfocado a un público más profesional con sus gestor de correo electrónico, agenda, notas etc. Una de sus mayores desventajas es lo difícil que es desarrollar una aplicación ya que se necesita una cuenta de desarrollador de RIM y estas suelen ser un poco costosas.

Y por último, el gran debutante en esta carrera y del que podemos esperara mucho es Firefox OS, el nuevo sistema operativo móvil de Mozilla. Su estrategia de mercado es fácil y muy coherente, entra al mercado para un público que apenas se está familiarizando con los Smartphones, un público joven, pero también está enfocado para los desarrolladores. Una ventaja de este sistema operativo es que los desarrolladores podrán ver esta primera versión de Firefox OS como una herramienta para poder desarrollar en HTML5. Una ventaja además de ser libre y abierto es que permitirá al público en general desarrollar y comercializar sus propias aplicaciones e incluso crear su propio modelo de negocio dentro de Firefox OS. Otra ventaja serán las aplicaciones de video llamadas basadas en la web, algo que será nuevo en la telefonía móvil. Algunas de las desventajas es que al ser totalmente nuevo tendrá un par de fallas, también en tema de seguridad por ser libre y abierto y también se pondrá a prueba el potencial de HTML5 en la web móvil.

A mi parecer Firefox OS revolucionara el mundo de las tecnologías móviles.

11 Bibliografía

- [1] Balaji, T.S. et al. (2005). Creating a Mobile Multimedia Service. *Communications of the ACM*, 48(7), p. 49 - 53.
- [2] HTML5, (n.d.) HTML5 Video, HTML5 Audio, obtenido de, <http://www.w3schools.com>
- [3] Douglas Crockford, JavaScript: The Good Parts. 2008, O'Reilly Media.
- [4] David Flanagan, JavaScript: The Definitive Guide, Sixth Edition. 2011. O'Reilly Media
- [5] DOM Scripting: Web Design with JavaScript and the Document Object Model: Second Edition. 2010
- [6] Julie Meloni, Sams Teach Yourself HTML, CSS, and JavaScript All in One. 2011, Sams.
- [7] "The New Generation of Web Applications" **Djamal Benslimane** *Lyon University*, **Schahram Dustdar** *Vienna University of Technology*, **Amit Sheth** *Wright State University*., p. 13 - 15.
- [8] "Predators and Prey" A new Ecology of Competition" by James F. Moore, Harvard Business Review
- [9] Tutorial "El lenguaje Objective-C para programadores C++ y Java" Fernando López Hernández, MacProgramadores.
- [10] sociable.co/mobile
- [11] <http://docs.blackberry.com/en/developers/>
- [12] Blackberry developer (n.d) What's a WebWorks App? Obtenida de, <http://developer.blackberry.com/html5/documentation>
- [13] Blogs, (12 Julio 2012) Desarrollo de Aplicaciones para Windows Phone (Herramientas), obtenido de, <http://blogs.msdn.com/b/ricardoj/archive/2012/07/12/desarrollo-de-aplicaciones-para-windows-phone-herramientas.aspx>
- [14] Intriducing to Firefox, obtenido de, <http://en.flossmanuals.net/firefox/>
- [15] Firefox Marketplace (n.d) Intro to firefox OS, obtenido de, https://marketplace.firefox.com/developers/docs/firefox_os

[16] Mozilla Support, (n.d.) Que es el Firefox Marketplace?, obtenido de, <https://support.mozilla.org/es/home>

[17] Mozilla Developer, (n.d). Getting started with app development, obtenido de, https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS

[18] Mozilla Wiki, (n.d.) Boot to Gecko Architecture, obtenido de, <https://wiki.mozilla.org/B2G/Architecture>