



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

**Aplicación para móviles que notifique eventos de
servicios escolares y culturales de la carrera de
Ingeniería en Computación**

TRABAJO ESCRITO
EN LA MODALIDAD DE DESARROLLO DE UN
CASO PRÁCTICO
PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTAN:

OSCAR GARCIA RODRÍGUEZ
JUAN ANTONIO VILCHIS REYES

ASESOR:

M. en C. Jesús Hernández Cabrera



MÉXICO 2013



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice General

1. Antecedente para el desarrollo de aplicaciones para móviles.
 - 1.1.- Dispositivos móviles
 - 1.1.1.- Clasificación de los dispositivos móviles
 - 1.2.- Sistemas operativos de los dispositivos móviles
 - 1.3.-Tecnologías para el desarrollo de Aplicaciones Móviles.
2. Tecnología ASP MVC 4 para el desarrollo de aplicaciones móviles.
 - 2.1.-Que es ASP MVC4
 - 2.2 Arquitectura ASP MVC4
 - 2.2.-Herramientas y requisitos para desarrollar una aplicación móvil en ASP MVC4
3. Framework PhoneGap para el desarrollo de aplicaciones móviles para diferentes plataformas.
 - 3.1- Origen de phonegap
 - 3.2.- Compatibilidad con las plataformas móviles
 - 3.3.- Arquitectura del framework phoneGap
 - 3.4- Herramientas y requisitos para desarrollar una aplicación móvil en phonegap
4. Análisis y diseño de la aplicación para la difusión de eventos.
 - 4.1-Planteamiento y Analisis del sistema.
 - 4.1.1-Casos de uso

4.2-Diseño del sistema.

4.2.1 Diagrama de clases (arquitectura)

4.2.2 Diseño de interfaz

4.3.- Intercambio de información y almacenamiento de datos.

Diagrama del modelo E/R de la base de datos.

5. Desarrollo e implementación de la aplicación.

6. Evaluación y trabajo a futuro

Conclusiones.

Introducción.

Al observar el desarrollo tecnológico que ha experimentado la humanidad desde mediados del siglo XX hasta hoy, no cabe duda de que más que un avance se ha producido una verdadera revolución. El descubrimiento de la informática, su aplicación paulatina en todo tipo de áreas de conocimiento y de producción, así como su introducción en la mayoría de los aspectos de la población a través de todo tipo de componentes, ha cambiado a nuestra sociedad y economía más rápido que cualquier otro hecho o descubrimiento anterior.

La computadora es uno de los inventos que mejor resume la nueva situación tecnológica. Aparecieron primero como enormes y costosas máquinas que solamente estaban disponibles en importantes universidades o centros de investigación. Con la aparición de nuevas técnicas de fabricación, como los circuitos integrados, su tamaño, sus capacidades, y sobre todo precio, variaron de tal forma que se convirtieron en un producto de masas, como son la televisión o la radio. La aparición de Internet, y sobre todo su apertura al público general, determinaron de forma inequívoca la importancia de las computadoras en la vida social, laboral o académica de cualquier persona hasta el día de hoy.

Simultáneamente a la aparición de Internet como servicio abierto, a principios de la década de 1990, surgió otro medio de comunicación que, si bien era más antiguo, se reinventaba a sí mismo gracias a los cambios en su tradicional soporte: la telefonía móvil. El *boom* de Internet, junto al furor de la telefonía móvil, confirmaba que esta revolución tecnológica no sólo afectaba a la investigación o a la actividad económica, sino que implicaba un fenómeno sociológico donde la comunicación y el acceso a la información en cualquier lugar y momento eran sus pilares básicos.

La reducción del tamaño de los componentes y el aumento en sus prestaciones y capacidades permitió que los usuarios contaran con mayores aplicaciones o usos, de forma que a través de un teléfono móvil no sólo se podían hacer llamadas o

enviar SMS, sino que, además se podía tener acceso a Internet, incluso funciones añadidas como realizar fotografías o vídeos.

Otros dispositivos de similar tamaño, no directamente relacionados con la telefonía, surgieron y se hicieron tan populares como los primeros. Desde ese momento puede empezar a usarse el término genérico *dispositivo móvil*.

Los más utilizados y conocidos en la actualidad, son aquellos que ofrecen mayor variedad de aplicaciones multimedia y los que más posibilidades de evolución presentan en este sentido.

A causa de estos avances, la gran aceptación e interés que la sociedad ha demostrado, para adoptar estos dispositivos como instrumentos cotidianos. Los dispositivos móviles han llegado a representar una nueva oportunidad de acceso hacia los sistemas de información, por tal motivo, y debido a la contingencia de población escolar que la FES Aragón alberga en su matrícula, usualmente se presenta la falta de recursos para una comunicación óptima, efectiva y puntual entre la institución y el alumno, esto ocasiona que en algunas situaciones sean difíciles los procedimientos escolares y se presente una escasa difusión en las actividades culturales y deportivas.

En este trabajo se realizará una aplicación para dispositivos móviles con una nueva herramienta llamada MVC 4 y EL Framework Phonegap, la cual le permitirá medir los alcances que nos pueden otorgar estas herramientas tanto en un plan de piloto como ya en su totalidad y así poder lograr el objetivo de la elaboración de la aplicación, el cual es: mantener a la población estudiantil informada sobre sucesos como: calendario escolar, evaluación de profesores, inscripción, cursos extracurriculares, actividades deportivas y culturales que acontecen en el plante y de esta manera la Facultad de Estudios superiores Aragón se sumará al tren de la tecnología.

El proyecto planteado en este trabajo es un sistema con fines demostrativos para la opción de titulación en la modalidad de caso práctico, para ello utilizaremos algunas tecnologías emergentes para el desarrollo de aplicaciones para dispositivos móviles.

La aplicación tendrá como principal objetivo mantener informado a los estudiantes acerca de las actividades recientes su citadas en el plantel en todos los ámbitos, para ello es necesario que el usuario tenga instalada la aplicación en un dispositivo móvil, se encuentre registrado para poder tener acceso a la información, la cual se estará actualizando conforme esta se vaya presentando.

Objetivos

Objetivo General: Implementar un sistema que facilite la comunicación entre los servicios que brinda el plantel y el alumno de la fes Aragón, para ellos haremos uso de nuevas tecnologías como son MVC 4 Y Phonegap que hoy en día nos facilitan proveer de información a una web aplicación, encapsulada como aplicación nativa y a su vez poder establecer el alcance que nos brindan las tecnologías ya mencionadas. Para hacer posible el desarrollo de esta aplicación; es necesario llevar a cabo los siguientes aspectos:

- Conocer las principales características de MVC 4 y EL Framework Phonegap.
- Compilar toda la información posible sobre MVC 4 y EL Framework Phonegap a fin de conocer sus arquitecturas, sus componentes básicos, y cuál es su comportamiento al ejecutar las aplicaciones, documentando todos estos aspectos.
- Además, de averiguarse cuáles son las ventajas y las posibilidades reales que MVC 4 y EL Framework Phonegap ofrecen frente a otros sistemas de similar naturaleza.

Metodología

El desarrollo de este proyecto se encuentra dividido en seis capítulos, con los cuales se pretende no solo instruir sobre las tecnologías utilizadas para el desarrollo de la aplicación; sino también poner en contexto y resaltar la importancia y los beneficios que se podrán obtener gracias al uso de los dispositivos móviles.

En el primer capítulo trataremos un poco sobre los antecedentes para el desarrollo de aplicaciones para móviles para ello es necesario abarcar que son los dispositivos móviles, la clasificación, los sistemas operativos móviles más comunes y las tecnologías existentes para el desarrollo de aplicaciones.

Para el capítulo segundo investigaremos lo referente a la tecnología ASP MVC 4, motivo por el cual indagaremos en aspectos como: la arquitectura, compatibilidad y los requisitos que son necesarios para elaborar una aplicación.

En el tercer capítulo se documentará lo relacionado con el framework Phonegap, origen, compatibilidad, arquitectura y los requisitos para la utilización de dicho Framework.

El capítulo cuatro comprende el “Análisis y diseño de la aplicación para la difusión de eventos”, por lo cual es necesario realizar una descripción de los pasos o las actividades que deberán realizarse (casos de uso).

Para los capítulos siguientes detallaremos el procedimiento realizado en el desarrollo de nuestra aplicación, las herramientas y los elementos empleados, así como los requisitos necesarios para la utilización de la aplicación. Por último haremos un análisis de este proyecto con el cual se podrá definir el alcance logrado, el cumplimiento de objetivos planteados y el conocimiento adquirido derivado de la elaboración de este trabajo.

Capítulo I

Antecedente para el desarrollo de aplicaciones para móviles.

1.1 Dispositivos móviles

No existe un consenso claro a la hora de definir qué es realmente un dispositivo móvil y qué no lo es. Es frecuente que hoy en día este término se utilice para designar únicamente a ciertos modelos de teléfonos móviles con mayores o menores prestaciones. A pesar de ello, un dispositivo móvil no tiene por qué ceñirse solamente al ámbito telefónico.

Buscando ser más rigurosos, se podría denominar dispositivo móvil a todo dispositivo electrónico que cumple unas características muy básicas:

- Es de reducido tamaño, haciéndolo fácil de transportar.
- Cuenta con una cierta capacidad de computación y almacenamiento de datos.
- Incorpora elementos de E/S básicos (por lo general, pantalla y/o algún tipo de teclado).

Más allá de estas características comunes, los dispositivos móviles forman en la actualidad un grupo sumamente heterogéneo y pueden incorporar casi cualquier componente de hardware y software que amplía y diversifica su función inicial. El más frecuente sin duda es la conexión telefónica (incluyendo servicios como el envío de SMS, MMS, y acceso WAP) o la conexión a Internet.

Igualmente son habituales la cámara fotográfica y de vídeo, pantalla táctil, teclado QWERTY, receptor de radio, Bluetooth, conexión mediante infrarrojos, dispositivos de memoria extraíbles, localizador GPS, acelerómetro, etc. Desde el punto de

vista del software, pueden incorporar también un amplio abanico de aplicaciones tales como programas ofimáticos, reproductores de audio y vídeo, organizadores, videojuegos, navegadores web o clientes de correo, entre otros.

1.1.1 Clasificación de los dispositivos móviles

Al igual que ocurre a la hora de definir qué es un dispositivo móvil, la clasificación que se pueda hacer de estos dispositivos está sujeta a diferentes valoraciones y a veces no existe un acuerdo amplio para ubicar un dispositivo móvil en una determinada familia.

En la década de los 90, tras la aparición de estos primeros dispositivos, establecer clasificaciones más o menos rigurosas era posible debido a que cada dispositivo estaba claramente definido para una función determinada o para un público concreto. El aumento de las prestaciones y funcionalidades que en la actualidad puede ofrecer cualquier dispositivo móvil dificulta el poder agruparlo dentro de un conjunto determinado. Por ejemplo, un *Smartphone* representa una evolución de un teléfono móvil tradicional, esto es, su cometido es ofrecer comunicación telefónica; sin embargo, cuenta con otros servicios adicionales como la conexión a Internet y aplicaciones, servicios propios de una PC, cámara de fotos y de vídeo o la posibilidad de reproducir películas o videojuegos¹.

La clasificación que a aquí se propone utilizar como principal criterio la funcionalidad o servicio principal para la que ha sido diseñado bien el propio dispositivo móvil, o bien aquel dispositivo del que directamente procede y del que supone una evolución mejorada. Tal y como se deduce de todo lo anteriormente explicado, la pertenencia a una categoría concreta no implica en absoluto que el dispositivo no pueda ofrecer otras muchas características propias de otras categorías. Dicho lo anterior, los dispositivos móviles pueden ser clasificados en los siguientes grupos:

¹ Roberto Ramírez Vilque. (2011). Tecnología y Desarrollo en Dispositivos Móviles. España: UOC

- **Dispositivo de comunicación:** Un dispositivo de comunicación es aquel dispositivo móvil cuyo cometido principal es ofrecer una infraestructura de comunicación, principalmente telefónica. Estos dispositivos ofrecen además servicios como el envío de mensajes SMS y MMS, o acceso WAP. En esta categoría se incluiría el tradicional teléfono móvil, precursor indiscutible dentro de los dispositivos móviles, la Blackberry y el *Smartphone*, que amplía considerablemente las prestaciones del primero mediante pantalla táctil, conexión a Internet o la ejecución de aplicaciones (por ejemplo, iPhone o HTC G1).
- **Dispositivo de computación:** Los dispositivos de computación son aquellos dispositivos móviles que ofrecen mayores capacidades de procesamiento de datos y cuentan con una pantalla y teclado más cercanos a una PC. Dentro de este grupo encontramos a las PDA, muy populares a finales de los años 90 y que permitían al usuario disponer de un organizador mucho más completo que los ofrecidos por los teléfonos móviles del momento, e incluso en ocasiones la visualización de documentos o acceso a Internet. Por otro lado, dispositivo de computación también es una máquina portátil o *Laptop*, que dentro de los dispositivos móviles son sin duda los que mayores prestaciones hardware ofrecen (igualando o superando a los de sobremesa) pero también los que tienen, con diferencia, un mayor tamaño, peso y precio. Las calculadoras gráficas pueden ser igualmente incluidas en este grupo de dispositivos de computación.
- **Reproductor multimedia:** Un reproductor multimedia es aquel dispositivo móvil que ha sido específicamente diseñado para proporcionar al usuario la reproducción de uno o varios formatos de datos digitales, ya sea audio, vídeo o imágenes. Dentro de estos dispositivos encontramos reproductores de MP3, los DVD portátiles, los eBooks, y en los últimos años los reproductores multimedia de la

popular familia. iPod de Apple, que ofrecen tanto audio y como vídeo. Estos dispositivos son con frecuencia los de más reducido tamaño y, junto a los teléfonos móviles y *Smartphone*, los más extendidos.

- **Grabador multimedia:** Dentro de los dispositivos móviles, un grabador multimedia es aquel dispositivo que posibilita la grabación de datos en un determinado formato digital, principalmente de audio y vídeo. En esta categoría se hallan las cámaras fotográficas digitales o las cámaras de vídeo digital.
- **Consola portátil:** Una consola portátil es un dispositivo móvil cuya única función es la de proporcionar al usuario una plataforma de juego. Las consolas portátiles fueron, junto a los teléfonos, los primeros dispositivos móviles en convertirse en un producto de masas. Hoy en día representan un importantísimo volumen de ventas dada su gran aceptación en la sociedad y son objeto de auténticas guerras comerciales entre las principales compañías del sector. Algunos ejemplos de esta categoría son la Nintendo DS de Nintendo, o la PSP de Sony.

Con base a lo anterior nos damos cuentas como a través del tiempo y en base a una convergencia digital el hombre ha podido integrar audio, video y datos de información en un dispositivo móvil.

Su aparición ha permitido aumentar enormemente la capacidad de almacenar, procesar y transmitir información y por ende, aumenta la capacidad de comunicación. Ésta digitalización de la información produce dos modificaciones sustanciales al modelo de comunicación masiva: la individualización de los contenidos y la interacción en tiempo real. Lo que permite el retorno de la inversión.

Otras de sus características son: la movilidad y comodidad que permite la tecnología digital ya que se pueden realizar comunicaciones más segmentadas en función al momento y lugar de contacto.

A su vez, esta transformación tecnológica permite la multiplicación de las

funciones desde un mismo aparato, como por ejemplo, el celular, en donde es posible desde leer el diario, ver e-mails, filmar, hablar por teléfono, etc. Y esto, trae como consecuencia se modifican los mercados².

1.2.-Sistemas operativos para dispositivos móviles

El sistema operativo destinado a correr en un dispositivo móvil necesita ser fiable y Tener una gran estabilidad, ya que incidencias habituales y toleradas en computadoras personales como reinicios o caídas no tienen cabida en un dispositivo de estas características. Además, ha de adaptarse adecuadamente a las consabidas limitaciones de memoria y procesamiento de datos, proporcionando una ejecución exacta y excepcionalmente rápida al usuario.

Estos sistemas han de estar perfectamente probados y libres de errores antes de incorporarse definitivamente a la línea de producción. Las posibilidades que existen en una PC de realizar actualizaciones e incluso reinstalar mejores versiones del sistema para cubrir fallos o deficiencias son más limitadas en un dispositivo móvil.

Es posible incluso que un dispositivo de esta naturaleza deba estar funcionando ininterrumpidamente durante semanas e incluso meses antes de ser apagado y Reiniciado, a diferencia de lo que ocurre con una PC. El consumo de energía es otro tema muy delicado: es importante que el sistema operativo haga un uso más racional y provechoso posible de la batería, ya que esta es limitada y el usuario siempre exige una mayor autonomía.

Todos estos aspectos de los dispositivos móviles, entre otros muchos, han de ser tenidos en cuenta a la hora de desarrollar un sistema operativo competente en el

² Buenas Tareas. (01 Noviembre 2012). Evolución Dispositivos Móviles. 10 Noviembre 2013, de Buenas Tareas Sitio web: <http://www.buenastareas.com/ensayos/Evolucion-Dispositivos-Moviles/5878652.ht>

mercado, atractivo para los fabricantes y que permita al usuario sacar máximo provecho de su termina³.

A continuación explicaremos los sistemas que son más comerciales:

- Symbian
- Windows Mobile
- Iphone OS
- Android

Symbian

Symbian es un sistema operativo para dispositivos móviles desarrollado por Psion, Nokia, Motorola y Ericsson. El principal objetivo de estas compañías era el de crear un nuevo y compartido sistema operativo que estuviera perfectamente adaptado a los teléfonos móviles del momento, y fuese además capaz de competir con Palm OS y Windows Mobile. La primera versión de Symbian, basada en el sistema EPOC de Psion, se lanzó en 1998. Actualmente, el número de empresas vinculadas al proyecto ha crecido considerablemente, siendo la última versión lanzada la 9.3 en julio de 2006.

El acuerdo bajo el cual se desarrolló Symbian es bastante simple: Symbian Ltd. Desarrolla el sistema operativo Symbian, que incluye el microkernel, los controladores, el *middleware* y una considerable pila de protocolos de comunicación e interfaces de usuario muy básicas. Los desarrolladores que obtienen la licencia correspondiente para trabajar con Symbian implementan sus propias interfaces de usuario y conjuntos de aplicaciones según las necesidades de sus propios dispositivos. Esto permitió a Symbian posicionarse como un sistema operativo muy flexible, que tenía en cuenta los requisitos de la mayoría de los dispositivos fabricados y, al mismo tiempo, permitía un alto grado de diferenciación.

Desarrollar aplicaciones para Symbian es relativamente sencillo. No es necesario

³ Juan Felix Basterretche . (2007). Dispositivos Moviles. 11 Agosto 2013, de UNNE Sitio web:
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/tfbasterretche.pdf>

Aprender ningún lenguaje de programación nuevo porque permite utilizar lenguajes habituales como Java, C++, Visual Basic o Perl, entre otros, para desarrollar aplicaciones. Este hecho ha permitido que actualmente sean cientos de miles las aplicaciones y utilidades disponibles para Symbian⁴.

Windows Mobile

Windows Mobile es un sistema operativo diseñado por Microsoft y orientado a una gran variedad de dispositivos móviles. En realidad, Windows Mobile representa una particularización de otro gran sistema de Microsoft llamado Windows CE.

A principios de la década de los 90, cuando comenzaron a aparecer los primeros Dispositivos móviles, Microsoft tomó la decisión de crear un sistema operativo capaz de hacer frente al entonces recientemente lanzado por Apple, el sistema Newton MessagePad. Fruto de esta iniciativa surgió Pegasus, cuyo nombre comercial definitivo fue Windows Compact Edition, o Windows CE.

El objetivo principal que buscaba Microsoft era que el nuevo sistema fuera lo suficientemente flexible y adaptable para poder ser utilizados en un amplio abanico de dispositivos, cuyo única característica común es la de ser de reducido tamaño y tener, por tanto, una limitación obvia en sus recursos. Las características principales con las que cuenta Windows CE son las siguientes:

- Es un sistema modular, lo que permite que cada fabricante pueda seleccionar aquellas partes que le benefician más para su dispositivo.
- Contempla una considerable gama de recursos hardware: teclado, cámara, pantalla táctil, etc.
- Tiene un tamaño en memoria relativamente pequeño y bajo coste

⁴ Sergi Hernando Sánchez (2010) Symbian OS: para dispositivos móviles inalámbricos Barcelona, Universidad Politécnica de Catalunya.

computacional.

- Es capaz de trabajar con distintas familias de procesadores de 32 bits.
- Permite interactuar con otros dispositivos móviles.

Arquitectura de Windows Phone

En la arquitectura de la plataforma Windows Phone se pueden utilizar dos formas de código que son: Silverlight y XNA, ambos están incluidos en una plataforma de ejecución, (Framework Managed Code) donde es administrada y aislada, como se observa en la imagen⁵.

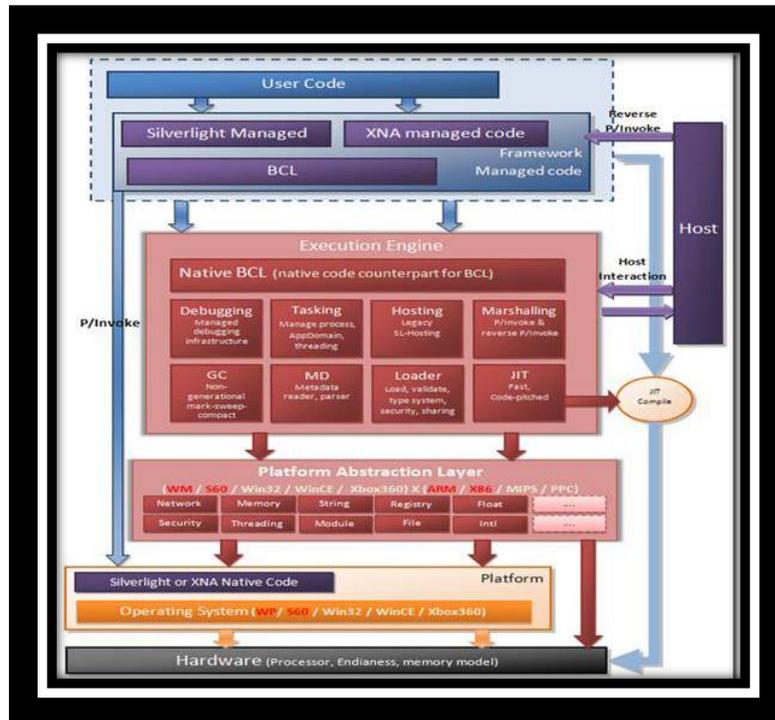


Figura1 Arquitectura de Windows Phone (Fuente:

<http://santimacnet.wordpress.com/2010/06/29/mi-primera-aplicacion-para-windows-phone-7/>)

⁵ César Tardáguila. (Enero 2006). Dispositivos Móviles y Multimedia.. 10 Agosto 2013, de UOC Sitio web: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9164/1/dispositivos_moviles_y_multimedia.pdf

ANDROID

Android constituye una pila de software pensada especialmente para dispositivos Móviles y que incluye tanto un sistema operativo, como *middleware* y diversas aplicaciones de usuario. Representa la primera incursión seria de Google en el mercado móvil y nace con la pretensión de extender su filosofía a dicho sector.

Todas las aplicaciones para Android se programan en lenguaje Java y son ejecutadas en una máquina virtual especialmente diseñada para esta plataforma, que ha sido bautizada con el nombre de Dalvik. El núcleo de Android está basado en Linux 2.6.

La licencia de distribución elegida para Android ha sido Apache 2.0, lo que lo convierte en software de libre distribución. A los desarrolladores se les proporciona de forma gratuita un SDK y la opción de unos plug-in para el entorno de desarrollo Eclipse varios que incluyen todas las API necesarias para la creación de aplicaciones, así como un emulador integrado para su ejecución. Existe además disponible una amplia documentación de respaldo para este SDK [5].

El proyecto Android está capitaneado por Google y un conglomerado de otras empresas tecnológicas agrupadas bajo el nombre de Open Handset Alliance(OHA). El objetivo principal de esta alianza empresarial (que incluye a fabricantes de dispositivos operadores, con firmas tan relevantes como Samsung, LG, Telefónica, Intel o TexasInstruments, entre otras muchas) es el desarrollo de estándares abiertos para la telefonía móvil como medida para incentivar su desarrollo y para mejorar la experiencia del usuario. La plataforma Android constituye su primera contribución en este sentido

·
Cuando en noviembre de 2007 Google anunció su irrupción en el mundo de la telefonía móvil a través de Android, muchos medios especializados catalogaron este novedoso producto como un nuevo sistema operativo, libre y específico para

teléfonos móviles. Sin embargo, los responsables del proyecto se han esforzado desde entonces en destacar que la motivación de Android es convertirse en algo más que un simple sistema operativo.

Con Android se busca reunir en una misma plataforma todos los elementos necesarios que permitan al desarrollador controlar y aprovechar al máximo cualquier funcionalidad ofrecida por un dispositivo móvil (llamadas, mensajes de texto, cámara, agenda de contactos, conexión Wi-Fi, Bluetooth, aplicaciones ofimáticas, videojuegos, etc.), así como poder crear aplicaciones que sean verdaderamente portables, reutilizables y de rápido desarrollo. En otras palabras, Android quiere mejorar y estandarizar el desarrollo de aplicaciones para cualquier dispositivo móvil y, por ende, acabar con la perjudicial fragmentación existente hoy día.

Además de todo ello, otro aspecto básico para entender la aparición de Android es que pretende facilitar la integración de estos dispositivos con las posibilidades cada día mayor ofrecidas por la Web. Por ejemplo, una aplicación desarrollada en Android, podría ser aquella que indicase al usuario, a través de Google Maps, la localización de sus diferentes contactos de la agenda y que avisase cuando éstos se encuentren a una distancia cercana o en una ubicación determinada.

Mejorar el desarrollo y enriquecer la experiencia del usuario se convierte, por tanto, en la gran filosofía de Android y en su principal objetivo⁶.

Arquitectura Android

La arquitectura de Android define estos cuatro elementos de los que podemos heredar:

⁶ Oscar Avila Mejia. (No citado). Android. 11 Diciembre 2013, de UAM Sitio web: <http://www.izt.uam.mx/newpage/contactos/revista/83/pdfs/android.pdf>

- **Activities:** Toda clase que como consecuencia de instanciarla implique una impresión por pantalla. Por ejemplo, en un programa lector de RSS sería la pantalla donde se listan los elementos nuevos.

- **Services:** Un servicio es un proceso que corre sin necesidad de utilizar una interfaz gráfica. En el ejemplo anterior, un servicio sería el proceso que se encarga de ir comprobando por periodos de tiempo estipulados, si hay o no algo nuevo en el RSS.

- **Intent:** Es la interpretación abstracta de una acción. Es semejante a un evento o a una interrupción. Como por ejemplo un clic, pulsación en pantalla o pulsación de un botón.

- **Content Providers & Broadcast Receivers:** Es uno de los mejores aspectos de Android y está enfocado a la reutilización de código en una aplicación, por tanto una aplicación puede tener ciertos elementos que sean llamados por cualquier otra aplicación para que realice una acción.

La diferencia entre Content Providers y Broadcast Receivers es que los primeros trabajan sobre URL's, (tipos de datos MIME) y los segundos trabajan a nivel de Intent. En el ejemplo que se explica anteriormente de RSS, uno sería llamado cuando se encontrara el valor "application/rss+xml" y otro cuando se lanza un "Intent".

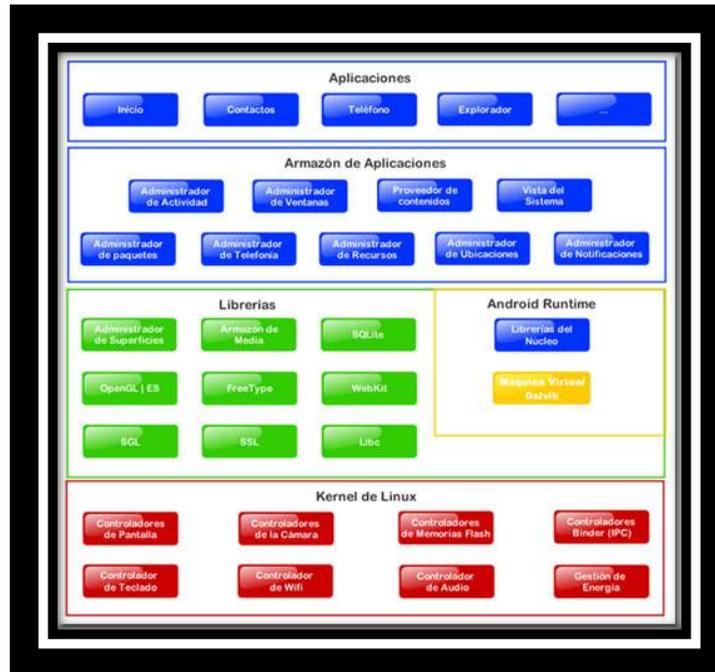


Figura 2 Arquitectura de Android (Fuente: Parrizas, Angel Alonso. Securely Deploying Android Devices.Pag.65)

IOS

Hablar de IOS es remonarnos al 9 de enero 2007 en la Macworld Conference & Expo, en donde apple presentaria el iphone. Aunque no fue hasta el 29 del mes junio cuando el lanzamiento del iOS. Era un sistema cerrado esto es que no presentaba aplicaciones nativas solo las que Apple había diseñado, y poco más: Mail, iPod, Calendar, Photos, Clock, Text, Safari, Notes, YouTube, Calculator, Maps, Settings, Camera, Stocks, and Phone. Aún no existía la App Store , realmente unb poco limitado, a contiucion vermeos como fue evolucionando.

iOS 2.0.

iOS 2.0 es hablar precisamente de la App Store, iTunes móvil y del iPhone 3G. El sistema venía con notificaciones push en el correo y varias pantallas de home, no hubo más funcionalidades introducidas, solo se facilitaba el acceso a aplicaciones de terceros.

iOS 3.0.

iOS 3.0. Las características más destacables de esa versión fueron las notificaciones push para advertir al usuario de los procesos que ocurrían en las

apps mientras estas estaban cerradas, la posibilidad de cortar, copiar y pegar, un compás y GPS mejoran la aplicación de mapas. El iPhone 3GS fue el primer teléfono de la compañía capaz de grabar video y enviar MMS.

iOS 4.

iOS 4.0 una versión que traía consigo los siguientes aspectos básicos: Las Multitareas, por fin había aplicaciones que se ejecutaban en segundo plano. Los Folders (carpetas) con la posibilidad de agrupar las aplicaciones como uno quisieran, una función simple pero útil. También se introduce FaceTime, iBooks, GameCenter.

iOS 5.

Siri, el humilde asistente personal, aparece en el iPhone 4S. Siri utiliza procesamiento del lenguaje natural para responder preguntas, hacer recomendaciones y realizar acciones mediante la delegación de las solicitudes a un conjunto de servicios web que va en aumento. Las afirmaciones del merchandising de Siri incluyen que ésta se adapta a las preferencias individuales del usuario con el paso del tiempo y personaliza los resultados, así como realizando tareas como reservar mesa para cenar o pedir un taxi.

iOS 6.

Lanzado apenas el 19 de septiembre 2013 iOS 6 elimina la aplicación de YouTube y presenta un sistema de mapas propio para reducir la interacción con Google. Los mapas incluyen vista en 3D y la App Store es remodelada. Ya se puede hacer llamadas de FaceTime a través de una red 3G (que antes estaba restringida a Wi-Fi) y el asistente Siri soporta nuevos idiomas como español e italiano(solo para iPhone 4s, iPhone 5 y iPod touch 5g). Passbook intenta sustituir los tickets y pases de vuelo. Y más de 200 novedades como Integración con Facebook, Email VIP etc.

Sin duda la evolucion de IOS hace que los demas sistemas operativos tengan que estar en cosntate renovacion.⁷

⁷ Desconocido. (21 Octubre 2011). Historia del software: iOS. 11 Agosto 2013, de Bit Sitio web: <http://bitelia.com/2011/10/historia-del-software-ios>

En el tema anterior describimos los sistemas operativos para móvil que hoy en día tienen más importancia en el mercado, esto lo han logrado por una parte a las aplicaciones que corren bajo sus plataformas, para poder crear dichas aplicaciones existen herramientas que nos permitirán o facilitar la creación de una aplicación que a continuación se exponen⁸.

1.3.-Tecnologías para el desarrollo de Aplicaciones Móviles.

Estudiado los principales sistemas operativos y lo que significan todo este conjunto para las aplicaciones móviles, veremos 2 tecnologías que han revolucionado la creación de aplicaciones y una de ellas será la que ocuparemos para el desarrollo del trabajo que presentamos.

Phonegap

Es un sistema para crear aplicaciones usando exclusivamente HTML5, CSS3 y Javascript, ejecutadas dentro en un componente WebKit del móvil. Provee una serie de librerías Javascript desarrolladas en el lenguaje específico de cada plataforma (Objective-C para IOS, Java para Android, etc) que nos permiten acceder a las características del móvil como GPS, acelerómetro, cámara, contactos, base de datos, filesystem, etc. Al ser una página web, tenemos acceso al DOM y podemos usar frameworks web como jQuery o cualquier otro. Requiere diseñar tu aplicación web con los componentes visuales típicos del HTML, etc o usar un framework web Mobile como jQuery Mobile o Sencha Touch entre otros. Tiene la ventaja de que puedes definir la navegación inicial de la aplicación usando el Chrome o el Firefox de tu ordenador, sin tener que ejecutarla en el simulador.

En resumen, podemos ver una aplicación PhoneGap como una serie de páginas web que están almacenadas y empaquetadas dentro de una aplicación móvil

⁸ Prof. Ing. Angel Caffa. (25 Julio 2008). Sistemas Operativos Para Dispositivos Móviles. 11 Diciembre 2013, de No citado Sitio
web:<http://miriammeza.files.wordpress.com/2010/09/presentacion.pdf>

visualizada con un navegador web, con acceso a la mayoría de Apis del móvil, lo cual lo convierte en una alternativa muy sencilla para crear aplicaciones.

Ventajas:

Es la solución que más plataformas móviles soporta, ya que corre dentro de un navegador web. Además de Iphone/Ipad y Android, funciona también en Palm, Symbian, WebOS, W7 y BlackBerry,

Es muy fácil de desarrollar y proporciona una gran libertad a los que tienen conocimientos de HTML y Javascript.

Hay buena documentación y bastantes ejemplos.

Es gratis, soporte de pago. Licencia BSD.

Inconvenientes:

Requiere Mac con Xcode para empaquetar aplicaciones IOS.

La aplicación no es más que una página web, por lo que el aspecto dependerá del framework web utilizado. Necesitaremos el uso de frameworks HTML móviles como Sencha Touch, jQuery mobile, Jo, Sproutcore, XUI, jQTouch si queremos que parezca una aplicación nativa.

No llega al rendimiento de una aplicación nativa, pues el HTML, CSS y Javascript debe ser leído e interpretado por el engine del navegador cada vez arranca.

Titanium Appcelerator: JavaScript

Con Appcelerator es posible crear aplicaciones para Android, Iphone y, además, de escritorio, usando exclusivamente Javascript (el soporte para Blackberry está en fase beta). Para programar proporciona Titanium Studio, un IDE basado en Eclipse con el que crear los proyectos y editar los ficheros Javascript y el resto de recursos y lanzar los scripts de creación. Experimentalmente, permite usar Php,

Ruby y Python también, pero transformando el código después en Javascript con los frameworks Phpjs, Skulpt o Ruby.js.

Aplicaciones móviles

Las aplicaciones se programan íntegramente con Javascript, creando y colocando “a mano” todos los controles, usando para ello una librería que hace de puente entre tu aplicación Javascript y los controles del sistema. Esto significa que las ventanas y demás controles visuales (botones, listas, menús, etc) son nativos: cuando se añade un botón, se crea un botón del sistema y se añade a la vista, lo que lo hace más rápido de renderizar y la respuesta del usuario es también la más rápida. A diferencia de PhoneGap, en Appcelerator no hay DOM, por lo que no se pueden usar librerías como jQuery que sirven para manipularlo, ya que el contexto de ejecución es Javascript puro, no dentro de un documento HTML.

Controles nativos y javascript compilado

Una de las características más interesantes de Appcelerator (y que más confusión produce, debido a que se vende como un generador de aplicaciones nativas: the native advantage), es que al empaquetar la aplicación, el Javascript es transformado y compilado. Después, cuando se arranca la aplicación en el móvil, el código se ejecuta dentro de un engine Javascript, tal y como dice la documentación oficial, que será JavaScriptCore en IOS (el intérprete de Webkit, el motor de Safari y Chrome) y Mozilla Rhino en Android/Blackberry. Si fuera compilado y ejecutado exclusivamente como Objective-C, como mucha gente piensa erróneamente, no serían posibles las capacidades dinámicas que tiene el lenguaje Javascript en tiempo de ejecución. Para saber más sobre como Titanium “compila” Javascript y hace “aplicaciones nativas”.

El hecho de que el Javascript esté compilado y que los controles creados sean nativos, le hace tener el mejor rendimiento posible (teniendo en cuenta que es Javascript, claro) en comparación con PhoneGap o Adobe Air para móviles.

Maquetación

Con Appcelerator es complicado maquetar, pues no existe un HTML inicial donde añadir los controles, sino que hay que crear las ventanas y componentes “a mano” con Javascript. Aún así, se pueden añadir componentes WebKit donde introducir HTML, pero solo se recomienda hacer cuando es absolutamente necesario para mostrar texto con formato, etc (ver este post donde explican cuando usar y cuando no usar HTML). También es posible crear un único componente Webkit a pantalla completa, y trabajar de manera parecida a como lo hace PhoneGap.

Por otro lado, puede que algo que se desarrolle para Iphone no funcione para Android, y viceversa. Los desarrollos de las librerías Javascript evolucionan por separado y hay que mirar bien qué se puede hacer y cómo. A diferencia de PhoneGap, que solo tiene una librería Javascript para acceder a las características especiales del sistema, Appcelerator necesita además librerías para manejar los controles nativos y su disposición en la pantalla, por lo que el desarrollo en general es más costoso.

Soporte IOS y Android

Sobre el soporte para Iphone/Ipad, es necesario tener Mac con Xcode 4 instalado, el entorno de desarrollo oficial de Apple para aplicaciones Mac y IOS, que es gratuito . Lo que hace Titanium Studio es generar un proyecto Xcode con el Javascript transformado junto con todas las librerías necesarias (tarda un poco, lo hace con unos scripts en Python que tiran del SDK de Iphone que tiene Xcode). Después es posible lanzar el simulador con la aplicación en Xcode sin salir de Titanium Studio. Una vez generado el proyecto, éste se puede abrir con Xcode y continuar empaquetándolo y configurándolo para su distribución (certificados, provisioning, logos, splash screen, etc). Desde Xcode no se puede editar el Javascript, pues para eso habría que volver a Titanium Studio y regenerar el proyecto Xcode otra vez. Seguramente sea posible que Titanium Studio ejecute o genere la aplicación en IOS directamente mediante scripts, sin usar Xcode, pero no lo he conseguido (hay dos opciones “Run iOS Device” y “Distribute App Store”)

Sobre el soporte Android, tanto para probar en el simulador como para empaquetar la aplicación, solo hay que tener el SDK de Android instalado.

Soporte de escritorio: Windows, Mac y Linux

En este caso Appcelerator funciona de manera parecida a como lo hace PhoneGap. Las aplicaciones se crean con HTML, CSS3 y Javascript, por lo que se puede usar jQuery o cualquier otra librería Javascript (ahora sí que existe DOM). De hecho, la aplicación es un WebKit, al igual que PhoneGap. El único problema grave que he visto es que cuando se empaqueta y distribuye la aplicación, todos los htmls, js y demás archivos aparecen en claro en la carpeta de la aplicación una vez instalada y pueden ser vistos y modificados. Otra pega es que es necesario hacer la exportación a cada sistema operativo desde ese mismo sistema operativo. Es decir, que para crear una aplicación final para Windows se debe hacer desde Windows, y que para crear una aplicación Mac, se debe hacer desde Mac.

Podemos ver y probar una aplicación de ejemplo para escritorio y para móviles. Se trata de un cliente de Twitter llamado Tweetanium, aquí tenemos el código fuente y descargas para todas las plataformas, y aquí la página de la aplicación en la App Store de Mac. Otra aplicación interesante creada con Appcelerator Wunderlist. Y en showcase de la página oficial hay muchas más.

Ventajas:

- Multiplataforma móvil y también de escritorio.
- Aspecto y controles nativos. El mejor rendimiento.
- Buenos ejemplos (el KitchenSink tiene prácticamente todo lo que se puede hacer)
- Gratis, soporte de pago. Licencia Apache.

Desventajas:

- Requiere Mac y Xcode para empaquetar aplicaciones IOS.

- Definición de componentes visuales y controles “a mano” (PhoneGap es HTML y Flex es MXML)
- Mucha documentación pero poco actualizada y desorganizada, tutoriales desfasados, poco profesional (en mi opinión)
- El IDE y las aplicaciones fallan a veces, no he podido exportar ni arrancar una aplicación Iphone desde el IDE, he tenido que hacerlo desde Xcode.
- Las aplicaciones de escritorio se distribuyen con el código fuente en claro (html, js, css, imágenes, todo).

Capítulo II

Tecnología ASP MVC 4 para el desarrollo de aplicaciones móviles.

2.1.-Que es ASP MVC4

Para poder entender el trabajo que exponemos es necesario hablar un poco de la historia de *MVC* y su evolución a través de las necesidades de las comunidades de desarrollo en torno al *framework.NET*

.

Una de las necesidades que se creó luego de que la web se convirtiera en un medio de comunicación masivo, fue la creación de un *framework* que permitiera el desarrollo de aplicaciones para la web; la respuesta de *Microsoft* a esta necesidad fue crear *ASP* o *Active Server Pages*, el cual es un lenguaje de *scripting* que permite la creación de sitios web en donde el código *ASP* y el código de presentación *HTML* se combinan en un único archivo para crear una única página.

En respuesta a esta primera forma de crear la web, surgió la necesidad de crear un *framework* que permitiera un proceso de desarrollo más transparente y montable, razón por la cual en el 2002 *Microsoft* lanza su librería de *ASP.NET* en respuesta a las necesidades de desarrollo web en aquella época. Es así como surgen los dos grandes grupos de desarrollo web en la plataforma de *Microsoft*: *Web Forms* y *ASP.NET MVC*. *Web Forms* responde a la necesidad de separar la capa de presentación y la capa lógica, permitiendo asignar a cada capa una responsabilidad única, además agrega un método que permite superar la

imposibilidad de guardar el estado del protocolo *HTTP*, de esta forma permite al desarrollador trabajar con estados en las aplicaciones web.

ASP.NET MVC al contrario de *Web Forms* aún deja la responsabilidad del manejo de estados al desarrollador, y se enfoca en crear una capa más que se encargue de manejar los modelos de datos, permite una separación efectiva de responsabilidades para cada capa.

En 2008 se lanza la primera versión de *MVC* como una alternativa de desarrollo web sobre la plataforma *ASP.NET*⁹.

¿Qué tiene de nuevo MVC4 en relación a sus versiones anteriores?

Entre las novedades que incluye MVC 4 en relación a su predecesor tenemos:

- Entity Framework 4.3, cuya principal ventaja es la migración de bases de datos usando únicamente código preservando los datos de la base de datos.
- Las aplicaciones se cargan de forma más rápida y cercanas para los usuarios, pues se disminuye el tiempo y el tamaño que hacen las páginas a los HTTP.
- Cache Busting que permiten un mejor manejo de archivos.
- “Wep API’s”: Se incluye soporte para su creación, permitiendo crear servicios HTTP y API’s que pueden ser llamadas directamente desde el código desde una gran cantidad de clientes. Ideal para crear Servicios REST.
- Web para móviles: ASP.NET MVC 4 incluye soporte para construir aplicaciones y sitios web para móviles, permitiendo crear de forma mucho más sencilla experiencias dirigidas a teléfonos y tabletas.

⁹ Rick Anderson. (17 de octubre 2013). Primeros pasos con ASP.NET MVC 5. 11 de diciembre 2013, de Microsoft Sitio web: <http://www.asp.net/mvc/tutorials/mvc-5/introduction/getting-started>

- Mejoras en Razor: ASP.NET MVC 4 incluye la versión 2 del motor Razor, este incluye mejoras en la creación de plantillas, más limpias y consistentes, soporte en la resolución de referencias URL, y en la muestra selectiva de atributos HTML¹⁰.

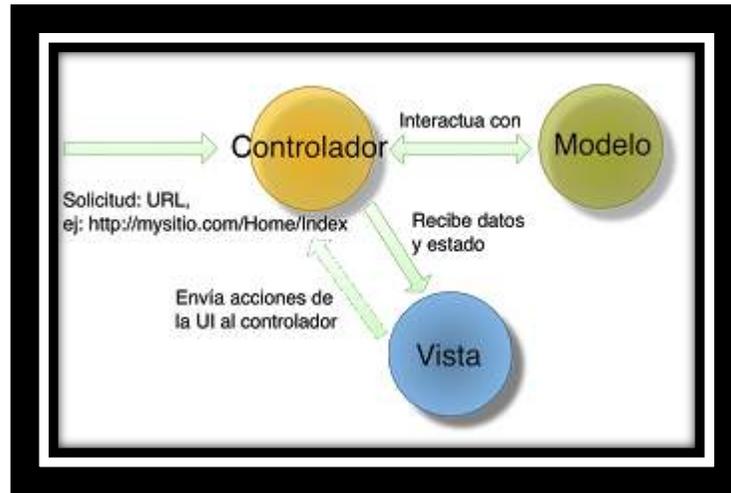


Figura 3 Arquitectura MVC4 Fuente: <http://www.asp.net/mvc/tutorials/mvc-5/introduction/getting-started>

Actualmente ASP.NET MVC 4 es compatible con Visual Studio 2010 y Visual Studio 2012, por lo que si desea seguir este tutorial y dispone de una versión anterior a las mencionadas tendrá que actualizar su versión de Visual Studio.

¿Qué ventajas tengo si lo uso?

La principal ventaja es la utilización del patrón MVC que permite organizar la aplicación de un modo más claro, donde cada parte tiene su responsabilidad, esto permite además, utilizar pruebas unitarias sobre el modelo si aplicamos bien el patrón, y también trabajar casi sin pensar en la presentación, vuelvo a aclarar, siempre que se utilice bien el patrón.

¹⁰ isamapro. (20 de febrero 2012). Lo nuevo en ASP.NET MVC 4. 11 de Noviembre 2013, de blog isamapro Sitio web: <http://ismapro.com/2012/02/20/lo-nuevo-en-asp-net-mvc-4-beta/>

Además tenemos total control de lo que ocurre, a diferencia de ASP.NET *webforms* en donde muchas cosas están fuera de nuestro control.

Por último es extremadamente extensible, lo que no hace el *framework* tal cual se ofrece podemos hacerlo con algún componente externo o hacerlo nosotros, esto incluye el manejo de rutas, los filtros, tipos de resultados, etc.

¿Qué desventajas tengo si lo uso?

La principal desventaja es que es más complejo lograr cosas que con ASP.NET con *webforms* son muy sencillas como por ejemplo conectar a una base de datos y mostrar en una grilla, la otra desventaja es que es necesario un mayor conocimiento de http para lograr aprovechar el potencial de la herramienta

2.2 Arquitectura ASP MVC4

El modelo arquitectónico Modelo-Vista-Controlador (MVC) separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. El marco de ASP.NET MVC proporciona una alternativa al modelo de formularios Web Forms de ASP.NET para crear aplicaciones web. El marco de ASP.NET MVC es un marco de presentación de poca complejidad y fácil de comprobar que (como las aplicaciones basadas en formularios Web Forms) se integra con las características de ASP.NET existentes, tales como páginas maestras y la autenticación basada en pertenencia. El marco de MVC se define en el ensamblado System.Web.Mvc¹¹.

¹¹ Mario Alberto Chavez . (18 Septiembre 2018). Programación Web: Patrón MVC. 01 Septiembre 2013, de Blog sobre desarrollo de software Sitio web: <http://mario-chavez.blogspot.mx/2008/09/programacin-web-patr-n-mvc.html>

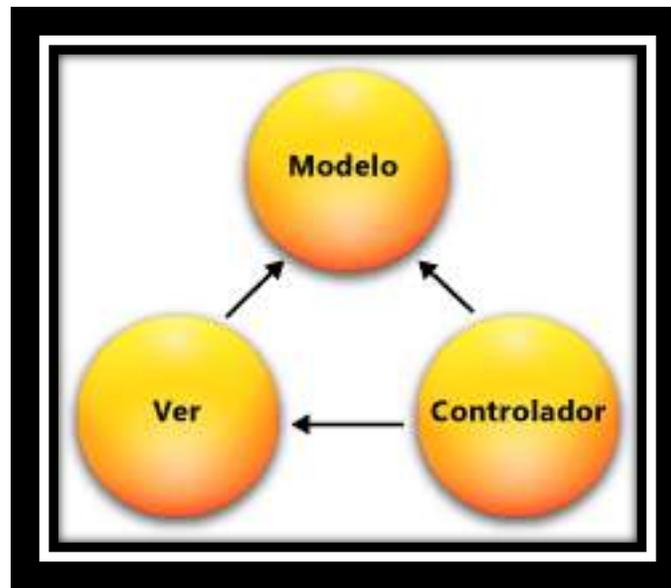


Figura 4 Modelo de diseño de MVC Fuente: <http://mario-chavez.blogspot.mx/2008/09/programacin-web-patrn-mvc.htm>

MVC es un patrón de diseño estándar con el que están familiarizados muchos desarrolladores. Algunos tipos de aplicaciones web salen beneficiados con el marco de MVC. Otras seguirán usando el modelo de la aplicación ASP.NET tradicional que está basado en formularios Web Forms y devoluciones. Otros tipos de aplicaciones web combinarán las dos estrategias; una no excluye a la otra.

El marco de MVC incluye los componentes siguientes:

- Modelos. Los objetos de modelo son las partes de la aplicación que implementan la lógica del dominio de datos de la aplicación. A menudo, los objetos de modelo recuperan y almacenan el estado del modelo en una base de datos. Por ejemplo, un objeto producto podría recuperar información de una base de datos, trabajar con ella y, a continuación, escribir la información actualizada en una tabla Productos de una base de datos de SQL Server.

En las aplicaciones pequeñas, el modelo es a menudo una separación conceptual en lugar de física. Por ejemplo, si la aplicación solo lee un conjunto de datos y lo envía a la vista, la aplicación no tiene un nivel de

modelo físico ni las clases asociadas. En ese caso, el conjunto de datos asume el rol de un objeto de modelo.

- Vistas. Las vistas son los componentes que muestra la interfaz de usuario de la aplicación. Normalmente, esta interfaz de usuario se crea a partir de los datos de modelo. Un ejemplo sería una vista de edición de una tabla Productos que muestra cuadros de texto, listas desplegables y casillas basándose en el estado actual de un objeto Producto.
- Controladores. Los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y por último seleccionan una vista para representar la interfaz de usuario. En una aplicación MVC, la vista solo muestra información; el controlador administra y responde a los datos proporcionados por el usuario y su interacción. Por ejemplo, el controlador administra los valores de la cadena de consulta y pasa estos valores al modelo, que a su vez podría usarlos para consultar la base de datos.

El modelo MVC le ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación (lógica de entrada, lógica de negocios y lógica de la interfaz de usuario), a la vez que proporciona un vago acoplamiento entre estos elementos. El modelo especifica dónde se debería encontrar cada tipo de lógica en la aplicación. La lógica de la interfaz de usuario pertenece a la vista. La lógica de entrada pertenece al controlador. La lógica de negocios pertenece al modelo. Esta separación le ayuda a administrar la complejidad al compilar una aplicación, ya que le permite centrarse en cada momento en un único aspecto de la implementación. Por ejemplo, se puede centrar en la vista sin estar condicionado por la lógica de negocios.

El acoplamiento vago entre los tres componentes principales de una aplicación MVC también favorece el desarrollo paralelo. Por ejemplo, un desarrollador de software puede trabajar en la vista, un segundo desarrollador puede ocuparse de

la lógica del controlador y un tercero se puede centrar en la lógica de negocios del modelo.¹²

2.2.-Herramientas y requisitos para desarrollar una aplicación móvil en ASP MVC4

Para empezar con el desarrollo con MVC 4 es necesario instalar las aplicaciones base del framework y un editor de código que permita un proceso de desarrollo ágil, como lo es Visual Studio; El siguiente proceso pertenece a la documentación de Introducción a ASP.NET MVC4, página 9

- Diríjase al sitio web <http://www.asp.net/mvc>
- Haga click en el botón de instalación en la página web
- Aparecerá una ventana emergente en donde se le solicita descargar el archivo VWDorVs11AzurePack.exe
- Una vez descargado en su sistema proceda a ejecutar el programa. A continuación aparecerá un recuadro con el nombre Web Platform Installer 4.0
- Dependiendo la configuración de su sistema el recuadro le mostrará las aplicaciones a instalar, las cuales son Visual Studio y Windows Azure.
- En caso de tener instalado Visual Studio 2012 el sistema realizará una actualización automática de librerías en un menor tiempo que la instalación típica.
- Haga click en Instalar.

¹² Microsoft ASP.NET Team. (27 enero 2009). MVC en ASP.NET. 15 agosto 2013, de Microsoft
Sitio web: <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>

- La aplicación se conectará automáticamente al sitio de Microsoft para descargar todos los archivos necesarios para la instalación. Tenga en cuenta que el tiempo de instalación depende directamente del ancho de banda de su red y la configuración del Equipo.
- Una vez instalado el sistema le presentará un resumen de todas las librerías y aplicaciones instaladas en el sistema
- Haga click en finalizar para terminar con la instalación.

Con la aplicación de desarrollo Visual Studio 2012, el framework ASP.NET y ASP.NET MVC 4, será posible crear el primer proyecto MVC 4.

Para crear el primer proyecto a continuación describiremos como hacerlo mediante los siguientes pasos que fueron tomados del manual de MVC y el cual lo ponemos tal cual para que no se pierda información¹³:

- Busque en su escritorio o en el menú de aplicaciones Visual Studio 2012 y ejecútelo.

- Si es la primera vez que ejecuta Visual Studio 2012, el sistema le preguntará qué tipo de configuración desea realizar para la edición, en este caso deberá seleccionar Desarrollo Web. Este procedimiento adecuará el editor para trabajar con toda la solución de Visual Studio orientada a desarrollo de aplicaciones web.

- Una vez haya completado el anterior proceso o en su defecto el editor se esté ya ejecutando, seleccione creación de nuevo proyecto.

¹³ Nelson Enrique Vera Parra & Octavio Jose Salcedo Parra & Steven Sierra Forero. (Desconocido). Introduccion a MVC 4. 11 Noviembre 2013, de Universidad Distrital Francisco José de Caldas Sitio web: <http://es.scribd.com/doc/129758998/Introduccion-a-MVC-4>

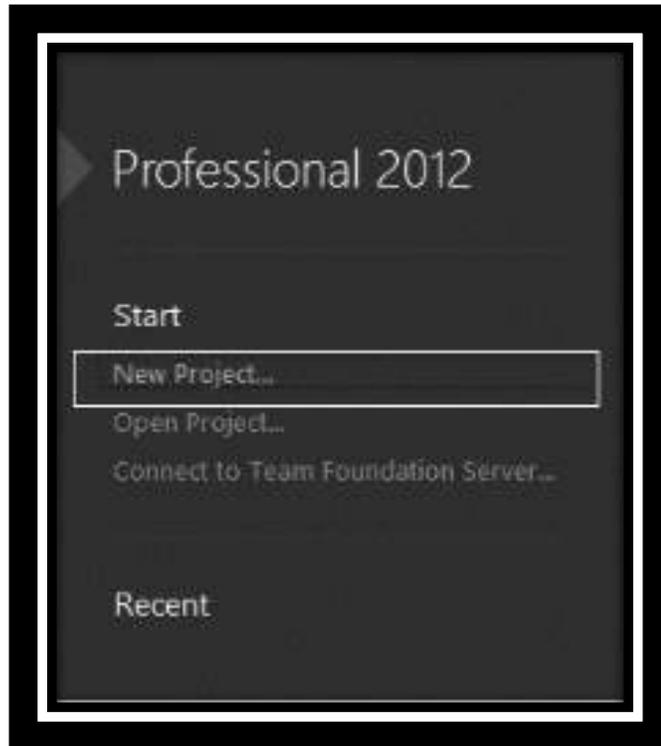


Figura 5 Menu creacion de proyectos de Visual Studio Fuente: <http://es.scribd.com/doc/129758998/Introduccion-a-MVC-4>

- Seleccione Visual C#, a continuación seleccione Web, allí encontrará diferentes opciones de las cuales deberá seleccionar ASP.NET MVC 4 Web Application.

Solo hay que seguir los pasos anteriores y tener conocimientos de cómo el entorno grafico de MVC y usted podrá realizar su primera aplicación.

PhoneGap es una propuesta de licencia OpenSource (código abierto), basada en JavaScript, que permite crear aplicaciones para dispositivos móviles utilizando HTML5 y CSS3. Esta nueva tecnología soporta el desarrollo de aplicaciones para los sistemas Android, IOS, BlackBerry, Windows Phone y Symbian.

Esta herramienta maneja APIs que permiten el acceso y control de elementos como el acelerómetro, cámara, contactos en el dispositivo, red, almacenamiento, notificaciones, entre otras.

PhoneGap cuenta con dos grandes ventajas al momento del desarrollo: se pueden ejecutar las aplicaciones en nuestro navegador web, sin depender de un simulador dedicado a esta tarea.

Lo que PhoneGap intenta es solucionar la complejidad a la que nos enfrentamos cuando queremos desarrollar una aplicación móvil y tenemos que decidir en qué plataforma vamos a trabajar, y por lo tanto que lenguaje de programación tendremos que aprender, porque las aplicaciones Android se desarrollan usando Java, iOS usa Objective C como lenguaje de desarrollo y Symbian usa C++.

Estamos entonces hablando de manejar muchos lenguajes de programación para desarrollar una sola aplicación para dispositivos móviles, eso sin mencionar todos los SDKs que debemos instalar para lograr esto.

Con PhoneGap hacemos un solo código cuya lógica de programación está sustentada en JavaScript, de esta forma no estamos obligados a aprender varios lenguajes de programación para poder migrar el código fuente que se hizo para una plataforma.

Podríamos definir entonces a PhoneGap como un framework para el desarrollo de aplicaciones nativas de sistemas operativos móviles, que hace uso de tecnologías web como HTML5, CSS3 y JavaScript.

‘

‘

Capitulo III

Framework PhoneGap para el desarrollo de aplicaciones móviles para diferentes plataformas.

3.1- Origen de phonegap

Para hablar del origen PhoneGap tenemos que remontarnos al PhoneDevCamp organizado por Apple en 2008 , en donde la empresa Nitobi Software presentaba un framework de código abierto que permite desarrollar aplicaciones multiplataforma usando HTML5, Javascript y CSS3.

Dicho framework da la posibilidad crea de las llamadas aplicaciones híbridas, a mitad de camino entre las nativas y las puramente web, esto lo consigue a partir de una serie de APIs que permiten controlar características de los dispositivos e interactuar con el hardware del móvil: Acelerómetro, cámara, contactos, eventos, geolocalización, redes o almacenamiento.

Las plataformas soportadas son: iPhone, Android, Blackberry, Windows Phone 7, Symbian, Bada y WebOS¹⁴.

¹⁴ Joan Anton Sances. (25 Febrero 2013). PhoneGap, una alternativa al código nativo. 11 Noviembre 2013, de Desconocida Sitio web: <http://blogs.evermind.com/TkQnO/component/content/?view=featured&start=5>

Por qué PhoneGap

Se debería utilizar PhoneGap para elaborar aplicaciones móviles por varias razones:

- Se ha desarrollado anteriormente una aplicación móvil utilizando tecnologías web y se quiere utilizar ésta en una o varias *app stores* (como Android Market, App Store o BlackBerry App World).
- Se quiere desarrollar una aplicación móvil a través de tecnología web y se necesita hacer uso de características del dispositivo (como la cámara o el calendario), que no están soportadas a través del navegador.
- Se quiere construir un prototipo rápido de una aplicación móvil sin utilizar Java o Objective-C.

Un gran número de aplicaciones comerciales disponibles hoy en día se desarrollaron utilizando PhoneGap; se puede encontrar una lista de algunas de ellas en www.phonegap.com/app. Este Framework se utiliza principalmente para aplicaciones de consumo (juegos, social media, utilidades, productividad), pero también se está utilizando en empresas para aplicaciones internas de cara a los empleados

¿Cuándo usar PhoneGap?

La versatilidad es uno de los puntos fuertes de PhoneGap, a continuación se analizará en qué ocasiones es recomendable usar este entorno de trabajo.

En aplicaciones informativas

Las aplicaciones informativas tienen como finalidad mostrar información estática y/o dinámica de productos, servicios, elementos multimedia (gráficos, textos, videos...)

En aplicaciones utilitarias

Son todas aquellas aplicaciones que nos sirven para hacer tareas con el móvil, desde la más crucial hasta la más trivial.

Las calculadoras, linternas, lectores de QR, etc. serían ejemplos de aplicaciones utilitarias.

En adaptación de Webapp

Una de las ventajas de PhoneGap es que se pueden crear aplicaciones completas usando los mismos lenguajes que se usan para crear sitios web: HTML5, CSS3 y Javascript.

Esto permite que con pequeñas adaptaciones se pueda adaptar una aplicación web a una aplicación nativa.

La aplicación de la Wikipedia es el mejor ejemplo. Dicha aplicación ofrece los mismos contenidos y servicios tanto en la aplicación nativa como en la web.

¿Cuáles son las principales ventajas de PhoneGap?

- 1- Programación: HTML5, CSS3 y javascript. Los programadores web se pueden adaptar fácilmente al mundo móvil mediante este framework.
- 2- Integración con frameworks de desarrollo web: JQuery Mobile, Sencha Touch, Dojo, jQTouch entre otros.
- 3- Rapidez. Con PhoneGap puedes realizar una aplicación sencilla en poco tiempo.
- 4- Multiplataforma: El mismo código fuente es usado en todas las plataformas.
- 5- Stores. Una vez finalizado el proyecto se puede empaquetar y subir fácilmente a las diferentes stores (AppleStore, PlayStore, etc.).
- 6- Facilidad de Uso. Existe multitud de formas de usar PhoneGap, con Adobe DreamWeaver (a partir de CS6) o con el plugin para XCode o Eclipse o directamente mediante la SDK.

7- Documentación. En su página web oficial, se puede encontrar una amplia sección de documentación con ejemplos.

8- PhoneGapBuild. Dispone de un potente compilador multiplataforma en la nube. Permite convertir la aplicación web en aplicaciones móviles (una para cada plataforma).

9- Moda. Solo hay que echar un vistazo a las redes sociales. PhoneGap es uno de los frameworks de moda.

3.2.- Compatibilidad con las plataformas móviles

- Sistemas operativos soportados:
 - Apple iOS
 - Nokia Symbian
 - Google Android
 - RIM BlackBerry
 - HP web OS
 - Samsung Bada
 - Microsoft Windows Phone

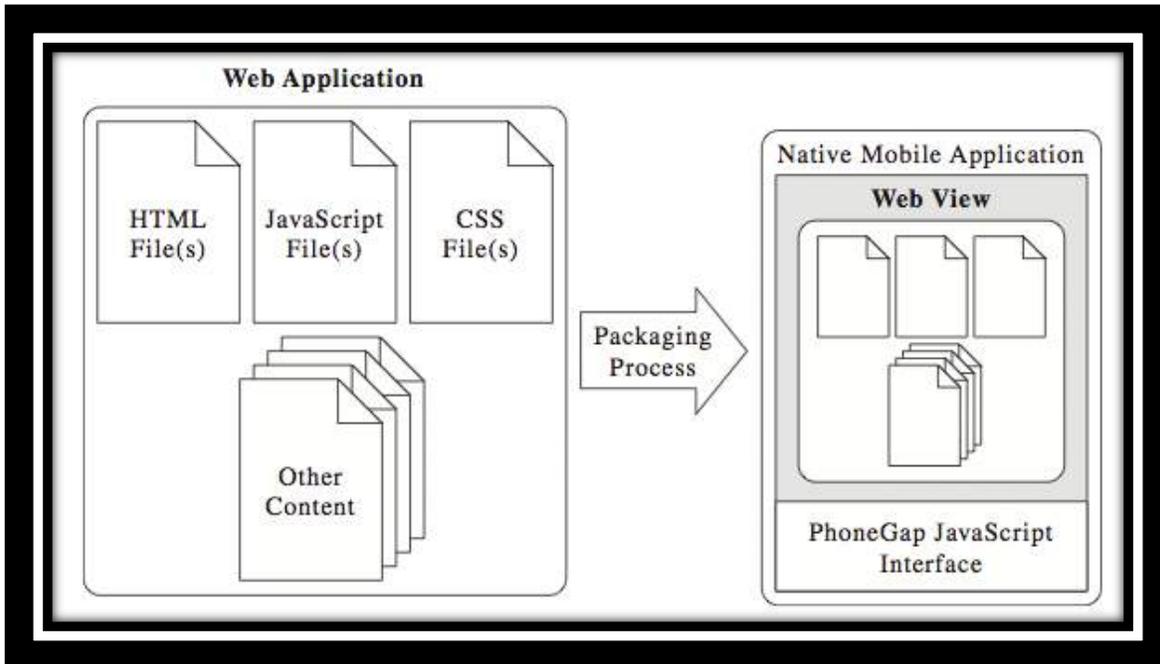


Figura 7 Funcionamiento de Phonegap Fuente: <http://phonegap.com>

Dentro de la aplicación nativa, la interfaz de usuario de la aplicación consiste esencialmente en una sola pantalla que no contiene más que una visión web única que consume todo el espacio disponible en la pantalla del dispositivo. Cuando la aplicación se inicia, se carga la página de inicio de la aplicación (normalmente `index.html`) en el navegador y, acto seguido, pasa el control al navegador para permitir al usuario interactuar con la aplicación. A medida que el usuario interactúa con el contenido de la aplicación, los enlaces o el código JavaScript dentro de la aplicación pueden cargar otros contenidos dentro de los archivos de recursos o pueden llegar a la red y extraer contenido de un servidor web o aplicación. Para algunas plataformas de dispositivos móviles como bada, Symbian y webOS, una aplicación nativa es una aplicación web; no hay concepto de aplicación nativa implementada en los dispositivos.

La aplicación web que se ejecuta dentro del recipiente o *container* es como cualquier otra aplicación web que se ejecuta en un navegador web móvil. Puede abrir otras páginas HTML; JavaScript incrustado en archivos de código fuente de la aplicación, ocultando o mostrando el contenido según sea necesario, reproduciendo archivos multimedia, abriendo nuevas páginas, realizando cálculos

3.3.- Arquitectura del framework phoneGap

El cliente phonegap se comunica con un servidor de aplicaciones para recibir datos. El servidor de aplicaciones maneja una lógica de negocio para comunicarse con una base de datos o repositorios.

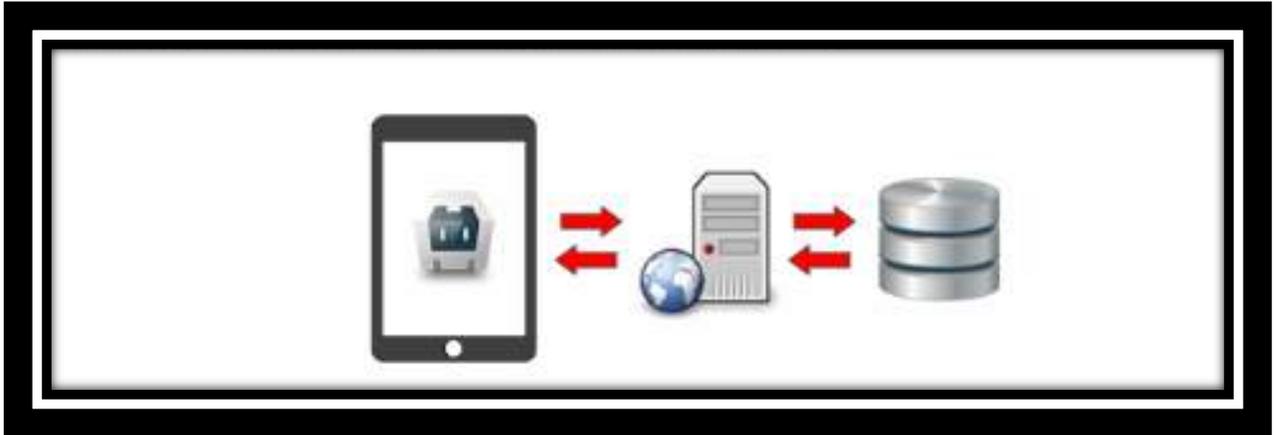


Figura 6 Arquitectura Phonegap Fuente: <http://phonegap.com>

El servidor de aplicaciones es normalmente un servidor web (apache, iis, etc...) con un lenguaje de scripting como coldfusion, java, .net. php.. Phonegap desconoce la tecnología que subyace y puede trabajar con un cualquier servidor de aplicaciones web.

3.4- Herramientas y requisitos para desarrollar una aplicación móvil en phonegap

PhoneGap nos permite crear aplicaciones nativas para dispositivos móviles (*Smartphone* y *tablets*) utilizando tecnologías web como HTML, CSS y JavaScript. Para ello se crea una aplicación web para el dispositivo móvil, y, gracias a las herramientas especiales que provee el paquete de PhoneGap, una aplicación nativa para cada plataforma móvil compatible.

y recuperando/mandando contenido al servidor. La apariencia de la aplicación está determinada por los atributos de fuentes, líneas, espacios, colores o sombras añadidos a elementos HTML o implementados a través de los CSS. Los elementos gráficos aplicados a las páginas pueden también ayudar a proporcionar un tema o un aspecto a la aplicación. Todo lo que uno pueda hacer en una aplicación web alojada en un servidor se puede hacer dentro de una aplicación PhoneGap.

Un navegador web móvil típico no tiene acceso a los componentes 'secundarios' del dispositivo como cualquiera de las otras aplicaciones que se ejecutan en dicho dispositivo (por ejemplo, los contactos), ni del hardware específico (acelerómetro, cámara, brújula, micrófono, etc.). La aplicación móvil nativa mítica, sin embargo, sí puede hacer uso frecuente de estos componentes. Para poder crear una aplicación móvil interesante, debe poderse acceder a dichos componentes del dispositivo fuera del navegador web. PhoneGap hace posible esta necesidad proporcionando un conjunto de APIs que él se pueden usar para permitir a una aplicación web que se ejecuta en el contenedor de aplicaciones de PhoneGap pueda acceder a los componentes del equipo que están fuera del contexto web. Como mencionamos anteriormente, una aplicación PhoneGap es un “nativo-wrapped” de aplicaciones web. Vamos a explorar cómo la aplicación web está “envuelta”.

Muchos SDK de desarrollo de móviles proporcionan un widget de navegador web (una “vista web”) como parte de su marco de interfaz de usuario (iOS y Android, por ejemplo). En aplicaciones puramente nativas, los controles Web de vista se utilizan para mostrar el contenido HTML ya sea desde un servidor remoto o local HTML empaquetado junto con la aplicación nativa de alguna manera. El nativo de “contenedor” aplicación generada por las cargas PhoneGap páginas HTML del desarrollador final de en uno de estos controles Web de vista y muestra el HTML resultante como la interfaz de usuario cuando se inicia la aplicación. Si los archivos JavaScript se incluyen en una página cargada en la vista web, este

código se evalúa en la página como normal. Sin embargo, la aplicación nativa que crea la vista web es capaz de (de diferentes maneras, dependiendo de la plataforma) de forma asincrónica comunicarse con código JavaScript que se ejecuta dentro de la vista web. Esta tecnología se conoce generalmente como “el puente” en el contexto de la arquitectura PhoneGap.

PhoneGap se aprovecha de esto para crear una API de JavaScript dentro de una vista web que es capaz de enviar mensajes y recibir mensajes de código nativo en la aplicación contenedora de forma asíncrona. La forma en que se implementa la capa de puente es diferente para cada plataforma, pero en iOS, al llamar a una lista de contactos, su invocación de método nativo entra en una cola de peticiones que se envían a través del puente. PhoneGap creará un iframe que carga un esquema URI (“brecha: / /”) que la aplicación nativa está configurado para manejar, momento en el que todos los comandos en cola será ejecutado. Comunicación de vuelta en la vista web se realiza mediante la evaluación de una serie de JavaScript en el contexto de la vista web de código nativo. Hay mucho más para PhoneGap que eso, pero los mensajes de la vista web en código nativo a través de la implementación del puente es la pieza clave de la tecnología que permite a las aplicaciones web locales para llamar a código nativo¹⁵.

¹⁵ Daniel Pérez. (25 Octubre 2012). PhoneGap. 12 Diciembre 2013, de Prezi Sitio web: http://prezi.com/kixvyzq7ola_/phonegap/

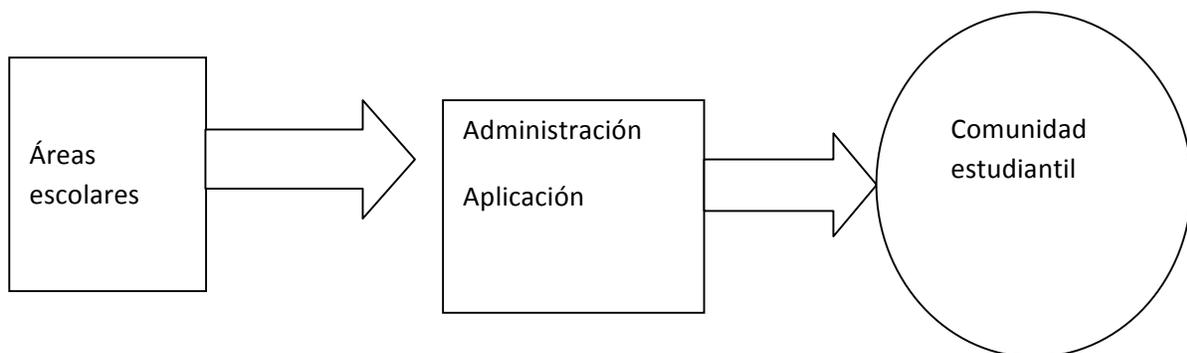
Capítulo IV

Análisis y diseño de la aplicación para la difusión de eventos.

En este capítulo se realizará el análisis necesario para el desarrollo de la aplicación. En la primera parte vamos a exponer la definición del problema y el análisis del sistema. Como parte central se realizará el análisis del diseño de la aplicación. Por último redactaremos la manera en que se realizará el intercambio de datos

4.1-Planteamiento y Análisis del sistema.

El objetivo de la aplicación es mantener al día al estudiante mediante noticias que él estará recibiendo en su móvil, para ello se tiene contemplado desarrollar dicha aplicación para los distintos sistemas operativos de móviles que actualmente tienen más auge en el mercado, en el siguiente diagrama se describirá el funcionamiento a gran escala de la aplicación.



En el primer recuadro observamos las distintas áreas que componen la escuela y las cuales tendrán como función proporcionar las noticias para la adecuada administración de la aplicación.

El siguiente recuadro es la aplicación, la cual estará compuesta con una interfaz donde el estudiante tendrá que registrarse para que así se vayan guardando los datos y se envíe la información correspondiente.

Por último el cual es el objetivo la comunidad estudiantil estará recibiendo los acontecimientos más importantes para su desarrollo académico.

4.1.1-Casos de uso

Los casos de uso son una técnica para especificar el comportamiento de un sistema:

“Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa cualquiera de sus servicios.”

Todo sistema de software ofrece a su entorno, una serie de servicios. Un caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software.

Un caso de uso es una colección de escenarios con un objetivo común:

—El conjunto de escenarios especifica un comportamiento que proporciona un resultado observable (típico) para uno o más actores u otros interesados en el sistema.

—Representa una tarea, o unidad coherente de funcionalidad, que el sistema está obligado a proporcionar a los actores en beneficio de los interesados.

—Puede tener asociado actores.

Un **escenario** es una secuencia de acciones que describe la funcionalidad del caso de uso.

—Escenario básico (todo va bien).

—Escenarios alternativos, manejo de errores, situaciones excepcionales.

Un **actor** es toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo.

En el caso de los seres humanos se pueden ver a los actores como definiciones de rol por lo que un mismo individuo puede corresponder a uno o más Actores. Suele suceder sin embargo, que es el sistema quien va a tener interés en el tiempo. Es frecuente encontrar que nuestros sistemas deben efectuar operaciones automáticas en determinados momentos; y siendo esto un requisito funcional obvio, resulta de interés desarrollar alguna forma de capturar dicho requisito en el modelo de caso de uso final.

En un caso de uso pueden participar varios actores distintos, y además:

—Las acciones de un actor pueden ser beneficiosas para otros actores.

—Puede haber interesados que no sean actores en absoluto.

Asociación: es la relación entre un actor y un caso de uso, o entre dos casos de uso. Este último caso se da cuando un caso de uso incluye a otro, extendiendo a otro o generaliza a otro.

Debido a que la función principal de un caso de uso es el describir el comportamiento de un sistema, esto mediante la especificación de las diversas maneras en las que los actores intervienen a través de los distintos escenarios previstos (pasos para realizar una acción).

Considerando lo vital que es la gestión de requisitos y basándonos en la técnica antes mencionada, con la finalidad de esclarecer lo mejor posible el proceso que se debe llevar a cabo para que los alumnos de la Facultad de Estudios Superiores Aragón puedan recibir notificaciones acerca de los eventos realizados por su escuela en sus dispositivos móviles, se elaboraron los siguientes casos de uso.

REGISTRO DE USUARIOS	
Actores:	Alumnos, Personal Administrativo.
Función:	Guardar la información de los usuarios.
Descripción:	Solicitar al usuario que proporcione su información personal para que pueda acceder a las noticias que se publicaran acerca de los eventos realizados por la Facultad de estudios Aragón.
Precondiciones:	El usuario no debe estar registrado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1.- Abrir la aplicación. 2.- Dar clic en el botón de Registro. 3.- Ingresar la información solicitada. <ul style="list-style-type: none"> ◦ Nombre ◦ Curp ◦ Semestre ◦ Carrera ◦ Número de Cuenta 4.- Dar clic en el botón de Guardar. 5.- Acceder a las noticias publicadas a cerca de los eventos organizados por la Facultad de Estudios Aragón.

Excepciones:	Se mostrara un mensaje a cualquier usuario que previamente haya realizado su registro.
Tablas:	<ul style="list-style-type: none"> ◦ tRegistro ◦ tUsuarios ◦ cCarrera ◦ cSemestre

INICIO DE SESIÓN	
Actores:	Alumnos, Personal Administrativo.
Función:	Permitir al usuario ingresar a la aplicación.
Descripción:	El usuario deberá proporcionar.
Precondiciones:	El usuario debe estar registrado en la aplicación.
Flujo Principal:	<p>1.- Abrir la aplicación.</p> <p>2.- Ingresar los datos necesarios.</p> <ul style="list-style-type: none"> ◦ Usuario ◦ Contraseña <p>3.- El usuario podrá visualizar los eventos organizados por la institución.</p>
Excepciones:	En caso de NO estar registrado se le indicara al usuario que realice el proceso de registro.
Tablas:	<ul style="list-style-type: none"> ◦ tRegistro ◦ tUsuarios

CAMBIO DE CONTRASEÑA	
Actores:	Alumnos, Profesores, Personal Administrativo.
Función:	Realizar el cambio de contraseña de los usuarios.
Descripción:	Los usuarios registrados podrán realizar el cambio de su contraseña cada vez que ellos lo crean conveniente.
Precondiciones:	El usuario debe estar registrado en el sistema y haber iniciado sesión.
Flujo Principal:	<ol style="list-style-type: none"> 1.- Abrir la aplicación. 2.- Iniciar sesión. 3.- Dar clic en botón Cambio de Contraseña. 4.- Ingresar la información necesaria para realizar el cambio. <ul style="list-style-type: none"> ◦ Contraseña anterior. ◦ Contraseña nueva. ◦ Confirmación de la nueva contraseña. 5.- La contraseña del usuario se ha modificado.
Excepciones:	Se enviara un mensaje en caso de que la contraseña nueva, no coincida con la confirmación.
Tablas:	<ul style="list-style-type: none"> ◦ tUsuarios

Visualizar Noticias	
Actores:	Alumnos, Personal Administrativo.
Función:	Guardar la información de los usuarios.
	Al ingresar a esta pantalla el usuario podrá visualizar en su

Descripción:	Al ingresar a esta pantalla el usuario podrá visualizar en su dispositivo móvil, todos los eventos organizados por la Facultad, dependiendo de la carrera y el semestre a los que se encuentre registrado.
Precondiciones:	El usuario debe estar registrado en el sistema. Haber iniciado sesión.
Flujo Principal:	1.- Ingresar a la aplicación. 2.- Iniciar sesión. 3.- Se mostrara al usuario la información correspondiente con su carrera y semestre a los que pertenece.
Excepciones:	Puede que no existan eventos registrados para los usuarios que correspondan con la información solicitada.
Tablas:	<ul style="list-style-type: none"> ◦ tRegistro ◦ tUsuarios ◦ cCarrera ◦ cSemestre ◦ tNoticias ◦ cTipo noticias

Registro de noticias	
Actores:	Alumnos, Personal Administrativo.
Función:	Dar de alta las nuevas noticias.
Descripción:	Para que los alumnos puedan enterarse con anticipación acerca de los eventos que se realizaran, es necesario que los usuarios con rol de administrador, registren las noticias acerca de los

Descripción:	Para que los alumnos puedan enterarse con anticipación acerca de los eventos que se realizaran, es necesario que los usuarios con rol de administrador, registren las noticias acerca de los nuevos eventos que se realizaran.
Precondiciones:	El usuario debe estar registrado en el sistema. Haber iniciado sesión. Tener un Rol de administrador.
Flujo Principal:	<ol style="list-style-type: none"> 1.- Ingresar a la aplicación. 2.- Iniciar sesión. 3.- Se mostrara al usuario toda la información que se haya cargado con anterioridad. 4.- El usuario podrá registrar los nuevos eventos a realizarse, así como modificar y/o eliminar registros anteriores. 5.- Una vez que se realizado el movimiento requerido, se deberá proceder a guardar los cambios realizados.
Excepciones:	Solo los usuarios con Rol de administrador podrán realizar acceder a esta funcionalidad.
Tablas:	<ul style="list-style-type: none"> ◦ tRegistro ◦ tUsuarios ◦ cCarrera

4.2-Diseño del sistema.

Dado que este proyecto pretende fundamentalmente el desarrollo de aplicaciones web, aunque finalmente las encapsulemos como aplicaciones nativas, usaremos la metodología MVC por ser la que mejor se adapta a las características de funcionamiento de las aplicaciones interactivas en general.

Este método, separa las capas de presentación de la aplicación, gestión de datos y dominio o negocio, en tres componentes: el modelo, la vista y el controlador. Básicamente su función será:

- **Modelo:** comprende las capas de gestión de datos y el dominio de la aplicación.
- **Vista:** es la capa de presentación, la interfaz con la que interactúa el usuario.
- **Controlador:** responde a las peticiones que realiza el usuario, generalmente mediante acciones y eventos de la interfaz. El controlador se comunica con el modelo y devuelve el resultado solicitado a la vista, pudiendo por tanto, producir cambios en ambos componentes.

El presente proyecto pretende como producto final, la creación de una aplicación interactiva, en forma de aplicación nativa para dispositivos móviles. Esto implica que será la interfaz de usuario el elemento central de su funcionamiento. Por tanto, el diseño de dicha interfaz, para cada uno de los tres módulos a desarrollar, se llevará a cabo bajo la metodología del diseño centrado en el usuario, cuyas etapas fundamentales son:

- **Análisis. Objetivos de la aplicación,** características de los usuarios, requerimientos del proyecto

- **Diseño. Diseño conceptual** (arquitectura de la información) y diseño visual
- **Prototipo. Prototipos de baja y de alta fidelidad.** Prototipo funcional
- **Evaluación de la usabilidad.** Testeo y pruebas de la interfaz

4.2.1 Diagrama de clases (arquitectura)

Una vez definida la aplicación que vamos a desarrollar y detallados sus contenidos y las metodologías implicadas, pasamos a describir la arquitectura en la que quedará enmarcada.

Tal como ya hemos comentado anteriormente, se trata de la creación de una aplicación web, la cual finalmente será encapsula y convertida en apps nativas para diferentes plataformas móviles. Por tanto, nos encontramos ante un clásico esquema usuario/servidor.

Analicemos sus componentes principales:

- **Servidor web**

La aplicación encargada de servir las páginas web creada para los alumnos. Además, en nuestro caso, nos servirá de alojamiento de usuarios.

Para ello, usaremos el servidor web IIS porque la aplicación nativa propuesta no consumirá muchos recursos

Servidor de bases de datos

Obviamente, la aplicación requerirá de una base de datos donde alojar la información de los alumnos registrados.

Hemos elegido el sistema de bases de datos relacionales SQL, por tratarse de software que es muy compatible en el caso que se presenta.

- **Clientes o Usuarios**

El proyecto comprende la creación de tres módulos independientes, cada uno con una funcionalidad determinada. Usaremos como usuario de las aplicación de gestión, cualquier navegador web compatible con las tecnologías HTML5.

Para la aplicación nativa que finalmente generaremos, nuestros usuarios serán los alumnos de ingeniería en computación que puedan acceder a través de dispositivos móviles Smartphone y tablet.

A continuación mostramos un esquema gráfico detallado, del funcionamiento de la arquitectura que acabamos de describir:

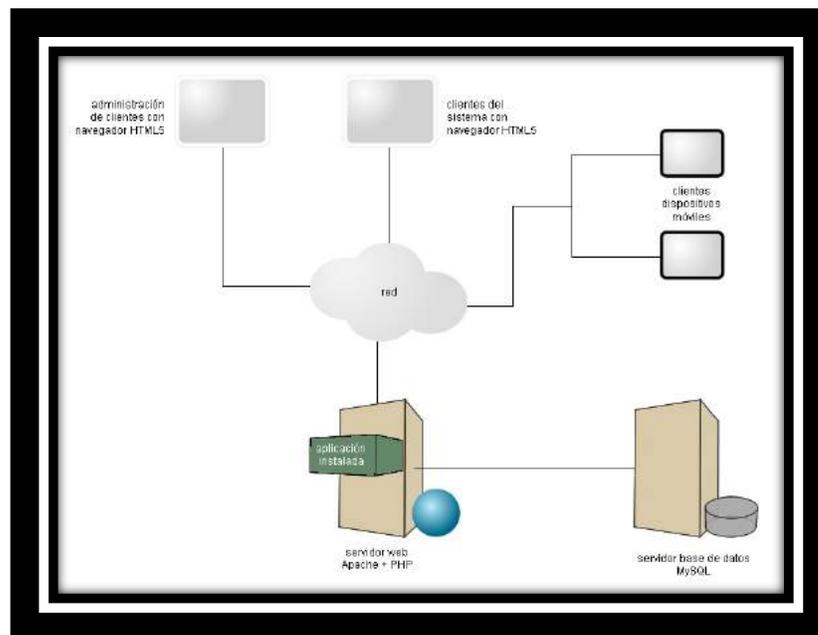


Figura 8 Arquitectura propuesta para la aplicación

4.2.2 Diseño de interfaz

El diseño de la interfaz fue planteada a través de un análisis el cual nos llevaría a determinar qué tipo de usuario tendría acceso al proyecto y los requerimientos.

Definido lo anterior decidimos que para no perder el estilo que ya tienen los portales de la UNAM ocuparemos parte del diseño que tienen algunas páginas para el proyecto, así que tomaremos el encabezado y pie, solamente cambiaremos el cuerpo para que la información que se envíe no se pierda.

Encabezado



Figura 9 Cabezera para el Desarrollo de la aplicación

Cuerpo o Fondo

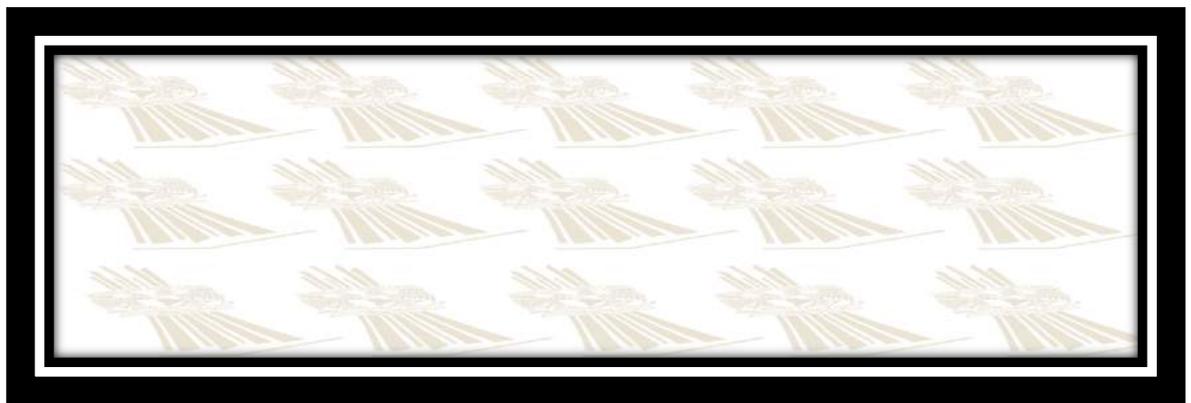


Figura 10 Fondo de la aplicación

Teniendo ya definido el esqueleto procedimos a indagar cual sería la manera más sencilla para el manejo de la app.

1.- Como la aplicación es para un servicio a los estudiantes lo más conveniente es saber si el alumno esta registrado o será la primera vez que utilizara la herramienta es por ello que para facilitar este proceso la interfaz solo contendrá dos opciones.

- Registro
- Iniciar sesión

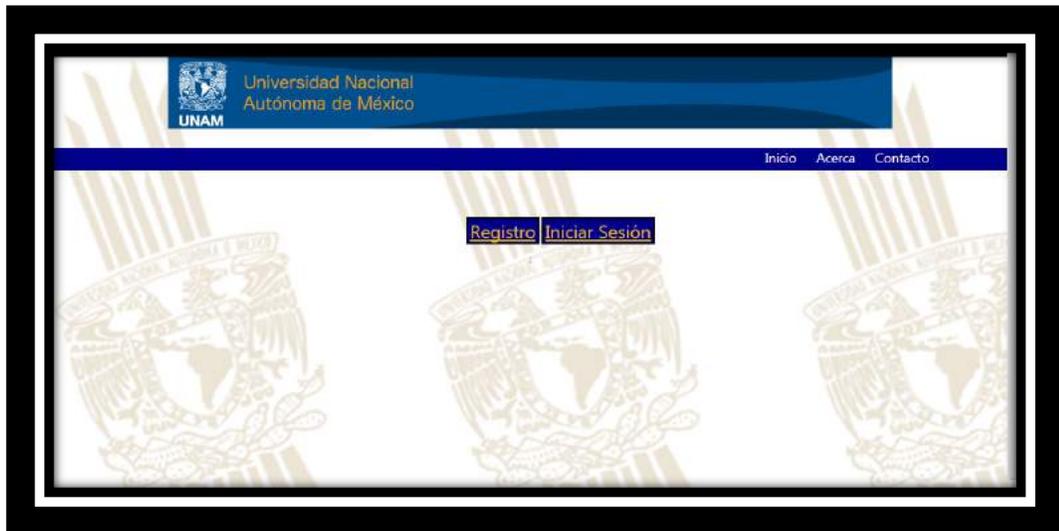


Figura 11 Interfaz de la aplicacion

Dentro de esta ventana podremos encontrar 2 opciones más que son:

- Acerca.- Se describe brevemente el objetivo de la carrera de Ingeniería en Computación.
- Contacto.- Los datos del personal de la jefatura de carrera



Figura 12 Interfaz acerca de la aplicacion

2.- Al elegir la opción de registro le aparecerá un formulario el cual se almacenara en la base de datos y la cual servirá para que el alumno ingrese a la herramienta.

Figura 13 Interfaz Registro Aplicacion

3.- La opción de iniciar sesión solo tendrá 2 campos los cuales son: usuario y contraseña, los cuales estarán almacenados en la base de datos.



Figura 14 Interfaz de Inicio de Sesion Aplicacion

4.3.- Intercambio de información y almacenamiento de datos.

Modelo E/R BD Diagrama

Una vez diseñado el esquema de funcionamiento de la lógica de la aplicación, procedemos al diseño y modelado de la base de datos de nuestro sistema. Como ya hemos indicado anteriormente, hemos optado por usar SQL Server 2008 R2 como gestor de bases de datos.

Del estudio y análisis de la lógica de la aplicación, concluimos la necesidad de implementar las siguientes tablas y campos, así como sus relaciones:

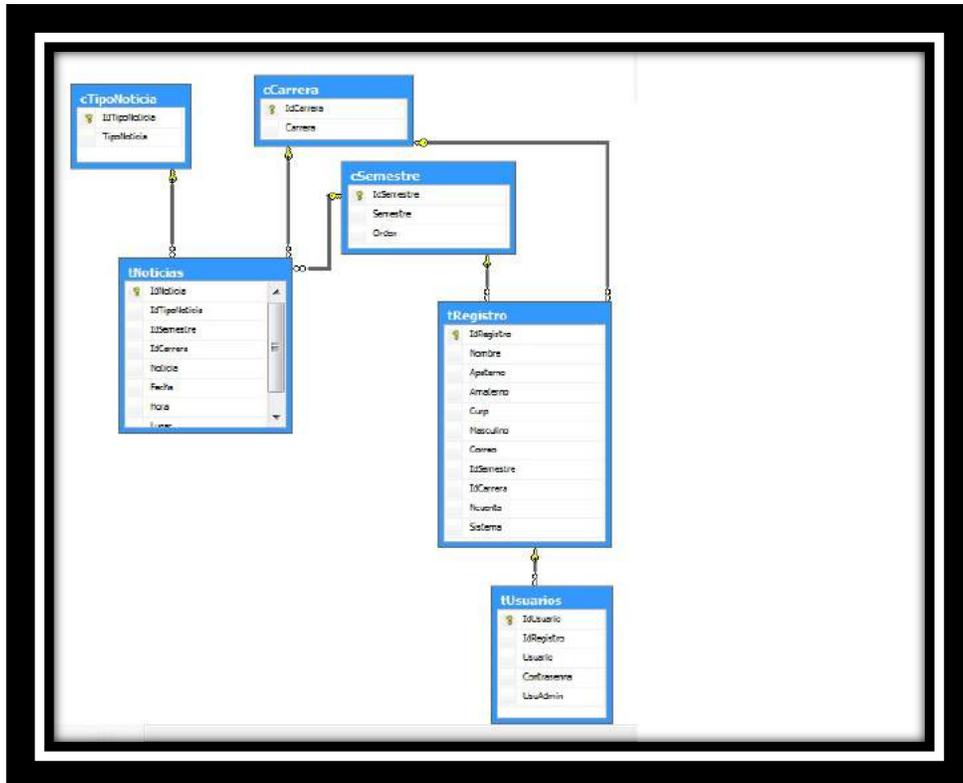


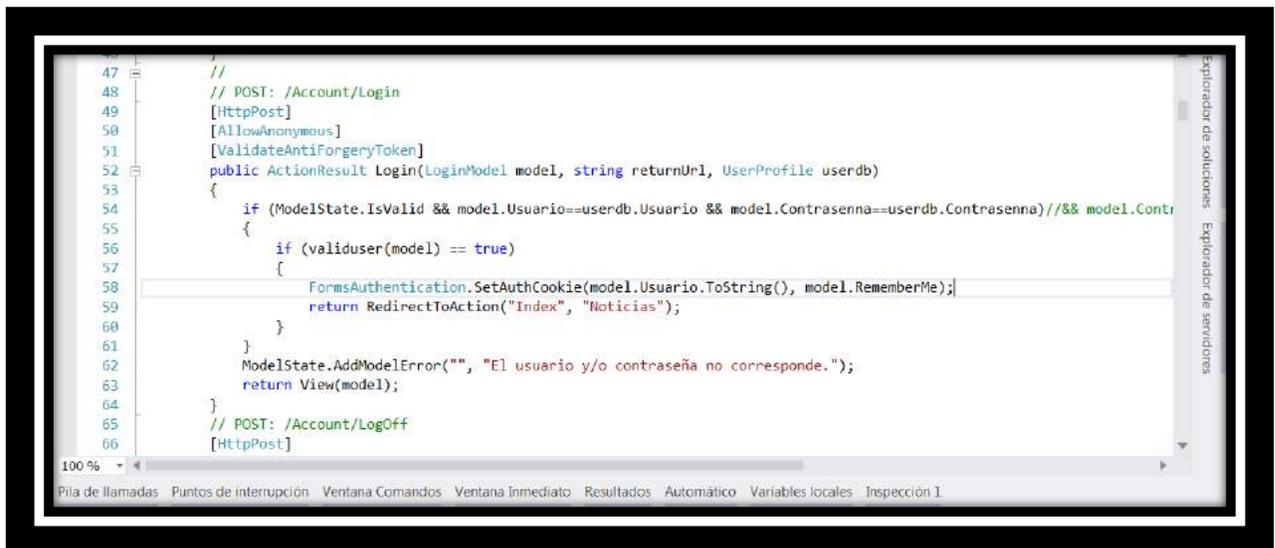
Figura 15 Diagrama del modelo E/R de la base de datos

Capítulo V

Desarrollo e implementación de la aplicación

En el desarrollo de la aplicación lo primero que se creó fue la parte lógica, el cómo se reconocería cada usuario, para esto se dividió en dos casos;

Caso 1,. El usuario ya está registrado



```
47 //
48 // POST: /Account/Login
49 [HttpPost]
50 [AllowAnonymous]
51 [ValidateAntiForgeryToken]
52 public ActionResult Login(LoginModel model, string returnUrl, UserProfile userdb)
53 {
54     if (ModelState.IsValid && model.Usuario==userdb.Usuario && model.Contrasenna==userdb.Contrasenna)//&& model.Contr
55     {
56         if (validuser(model) == true)
57         {
58             FormsAuthentication.SetAuthCookie(model.Usuario.ToString(), model.RememberMe);
59             return RedirectToAction("Index", "Noticias");
60         }
61     }
62     ModelState.AddModelError("", "El usuario y/o contraseña no corresponde.");
63     return View(model);
64 }
65 // POST: /Account/LogOff
66 [HttpPost]
```

Figura 16 Metodo para identificar que el usuario esta registrado

En la imagen anterior podemos observar la parte lógica, en donde si ya está registrado, teclea su usuario y contraseña para que el sistema le de acceso a las noticias, en caso de que no lo esté e intente hacer login se enviara un mensaje de usuario incorrecto

Caso 2 El usuario no está registrado

```

Controllers.AccountController Register()
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Register(RegisterModel model, String semestreList, String CarreraList, String GeneroList)
{
    if (ModelState.IsValid)
    {
        var _semestre = Request.Form["semestreList"];
        var _carrera = Request.Form["CarreraList"];
        var _genero = Request.Form["GeneroList"];
        var _sistema = 1; // escolarizado(1) o abierto (0)
        try
        {
            var db = new UsersContext();
            using (SqlConnection connection = new SqlConnection(db.Database.Connection.ConnectionString))
            {
                try
                {
                    SqlCommand command = new SqlCommand("UspRegistro", connection);
                    command.CommandType = CommandType.StoredProcedure;
                    command.CommandTimeout = 300;
                    command.Parameters.Add(new SqlParameter("@Nombre", SqlDbType.VarChar)).Value = model.nombre;
                    command.Parameters.Add(new SqlParameter("@Apaterno", SqlDbType.VarChar)).Value = model.Apaterno;
                    command.Parameters.Add(new SqlParameter("@Amaterno", SqlDbType.VarChar)).Value = model.Amaterno;
                    command.Parameters.Add(new SqlParameter("@Carp", SqlDbType.VarChar)).Value = model.Carp;
                    command.Parameters.Add(new SqlParameter("@Masculino", SqlDbType.VarChar)).Value = _genero;
                    command.Parameters.Add(new SqlParameter("@Correo", SqlDbType.VarChar)).Value = model.correo;
                    command.Parameters.Add(new SqlParameter("@IdSemestre", SqlDbType.VarChar)).Value = _semestre;
                    command.Parameters.Add(new SqlParameter("@IdCarrera", SqlDbType.VarChar)).Value = _carrera;
                    command.Parameters.Add(new SqlParameter("@Ncuenta", SqlDbType.VarChar)).Value = model.Cuenta;
                    command.Parameters.Add(new SqlParameter("@Sistema", SqlDbType.VarChar)).Value = _sistema;
                }
            }
        }
    }
}

```

Figura 17 Metodo para identificar que el usuario no esta registrado

Si el usuario es la primera vez que utiliza la aplicación se le pedirá que llene un combo para poder acceder a la herramienta.

Teniendo ya el funcionamiento y la interfaz deseada es hora de empaquetar para cada sistema operativo

Una vez que ya se termino la parte de interfaz y el funcionamiento de la aplicacion, surge la necesidad de cómo hacer para que nuestra aplicación nativa cumpliera con su función.

Para ello analizamos varias tecnologías, pero sin duda la más acoplada para este proyecto es IIS en servidor web del sistema operativo Windows.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Configuración e instalación de IIS

1.- Nos dirigimos a panel de control y elegimos la opción de programas veremos los programas que tenemos instalados

2.-En la parte izquierda veremos la opción de características de Windows, se abrirán las características de Windows, sin embargo a nosotros nos interesa internet information services, al localizar y desplegar las opciones ni una esta seleccionada, por lo que procederemos a seleccionar todas una por una

‘ 3.-Una vez seleccionado todo le damos en aceptar y nos aparecerá un mensaje en donde Windows realizara los cambios

4.-Una vez que se configuro, nos dirigimos a cualquier explorador y en ka barra de dirección ponemos local host y nos muestra la página de bienvenida, por lo cual ya está instalado

5.- Otra manera de saber si se configuro bien nuestro iis es ingresando al disco local y ahí encontraremos las carpetas que se muestran en la siguiente imagen

6.- Una vez que ya nos cercioramos que nuestro servidor está configurado, procedemos a publicar el proyecto de MVC,

7.- Ya publicado nos creará una serie de archivos en .rar, los cuales nos ayudaran para publicar la página web.

Empaquetamiento para android.

Abrimos eclipse—abrir —nuevo ---proyecto

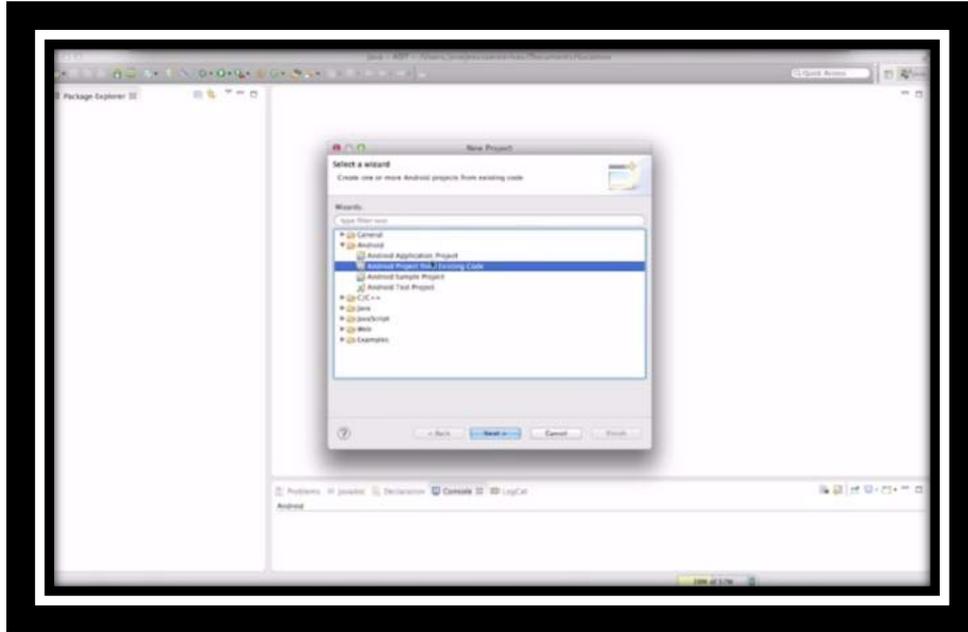


Figura 18 interfaz de eclipse para crear proyecto

En esta parte al pulsar abrir se incorpora el proyecto que ya tengamos listo.

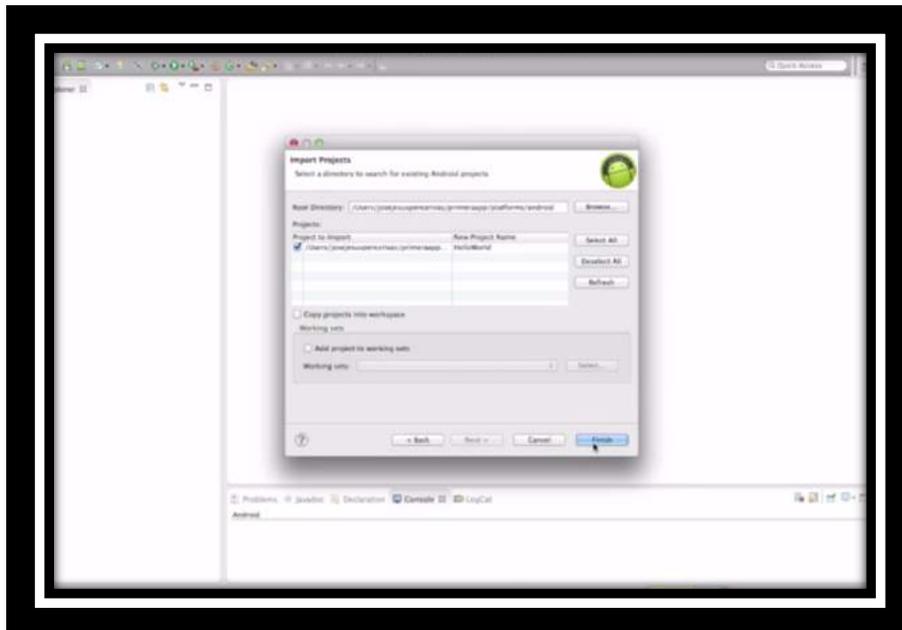


Figura 19 interfaz de eclipse para exportar proyecto

Reconoce el proyecto y añade todos los archivos dentro del explorador

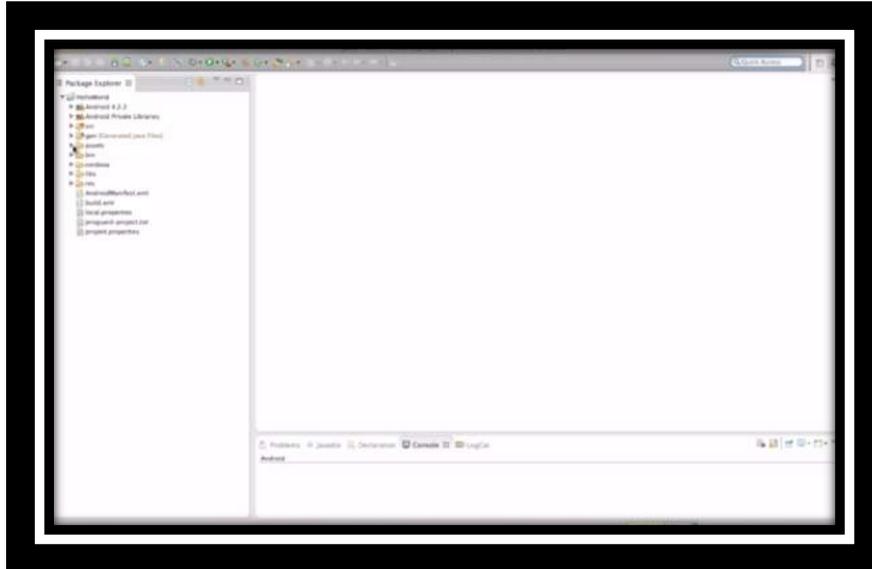


Figura 19 interfaz de eclipse para iniciar proyecto

Pulsamos el botón iniciar y aparecerá el emulador cargando la aplicación.

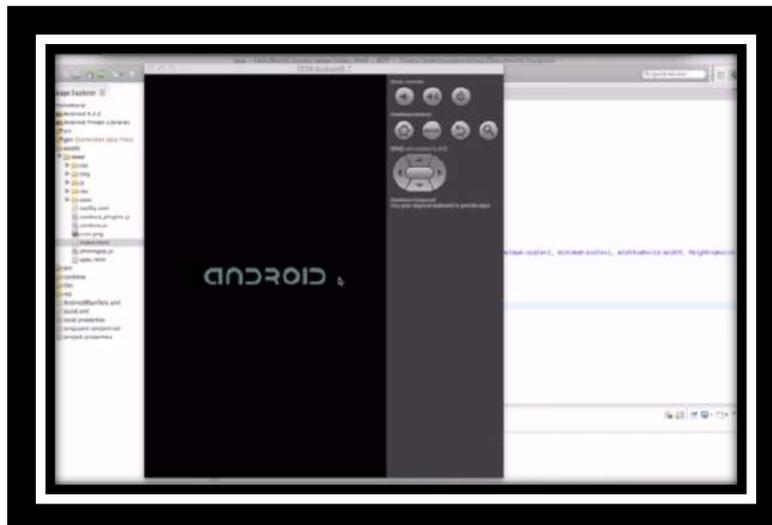


Figura 20 Emulador Android

Al finalizar la carga podemos ya observar la aplicación como un elemento más del sistema.



Figura 21 Aplicación Noticias en emulador

En nuestro caso solo emulamos la aplicación para ser más gráfico el proyecto escrito.

Empaquetamiento BlackBerry

Requisitos:

El entorno de desarrollo está disponible para Windows, Linux y Mac.

Instalar el SDK nativo de BlackBerry

El SDK nativo de BlackBerry está disponible en developer.blackberry.com. Después de la instalación, tienes que añadir sus herramientas de línea de comandos a la ruta del sistema.

En Windows:

- Ir a mis Variables de entorno informático → propiedades → avanzada →.
- Anexar el directorio de instalación del SDK nativo a la ruta, por ejemplo:

```
;C:\bbndk\host_10_2_0_132\darwin\x86\usr\bin\
```

En Mac y Linux:

- Editar el `~/.bash_profile` archivo, añadir una línea como la siguiente, dependiendo de donde se instaló el SDK nativo:

```
$export PATH = ${PATH} :/Applications/Momentics.app /host_10_2_0_15/Darwin /x86 /usr/bin/
```

- Ejecute lo siguiente para aplicar el cambio en el actual período de sesiones:

```
$ fuente ~/.bash_profile
```

Desplegar en emulador

Si desea ejecutar un emulador de dispositivo, descargue e instale el simulador de BlackBerry 10.

- Descargar
- Getting Started

Antes de probar una aplicación en un emulador o un dispositivo, tienes que activar el modo de desarrollo.

Inicie la imagen del emulador, luego elija la configuración de la pantalla de Inicio.



Figura 22 Emulador Black Berry

Empaquetamiento IOS

Crear un proyecto

Ejecute el comando crear, especificando la ruta existente para el proyecto, el identificador de paquete de reversa-dominio-estilo y nombre para mostrar de la aplicación.

```
$ ./path/to/cordova-ios/bin/create /dirección del nuevo proyecto a crear
com.example.nombre_proyecto Nombre a mostrar
```

Construir un proyecto

```
$ /dirección del proyecto creado /cordova/build
```

Requisitos y Soporte

Herramientas necesarias para construir aplicaciones de iOS sólo se ejecutan en el sistema operativo OS X en ordenadores Mac basados en Intel de Apple ®. Xcode ® 4.5 (la versión mínima necesaria) sólo se ejecuta en OS X versión 10.7 (Lion) o mayor, e incluye el iOS 6 SDK (Software Development Kit). Para enviar aplicaciones de la Apple App Store SM requiere las últimas versiones de las herramientas de Apple.

Usted puede probar muchos de las características de cordova utilizando el emulador IOS se instala con el SDK de iOS y Xcode, pero es necesario un dispositivo real para probar plenamente todas las características del dispositivo de la aplicación antes de enviarla a la App Store. El dispositivo debe tener al menos iOS 5.x instalado, la versión mínima soportada como iOS de Córdoba 2.3. Dispositivos de apoyo incluyen todos los modelos de iPad ®, iPhone ® 3GS y superiores, y iPod ® touch de 3^a generación o posterior. Para instalar aplicaciones en un dispositivo, también debe ser miembro de Apple iOS Developer Program , que cuesta \$ 99 por año. Esta guía muestra cómo implementar aplicaciones en el emulador de iOS, para el que no es necesario registrarse en el programa de desarrolladores.

Instale el SDK

Hay dos maneras de descargar Xcode:

- De la App Store , disponible mediante la búsqueda de "Xcode" en la App Store la aplicación.
- A partir de Apple Developer Descargas , lo que requiere el registro como desarrollador de Apple.

Una vez que está instalado Xcode, varias herramientas de línea de comandos necesitan ser habilitado Cordova se ejecute.Desde el Xcode menú, seleccione Preferencias , entonces el Descargas pestaña. Desde

el Componentes del panel, pulse el Instale botón junto a la línea de comandos Herramientas listado.

Implementar en el emulador

Para obtener una vista previa de la aplicación en el emulador de iOS:

1. Asegúrese de que el *xcodeproj.* se selecciona un archivo en el panel de la izquierda.
2. Seleccione la **hola** aplicación en el panel situado a la derecha.
3. Seleccione el dispositivo destinado a partir de la barra de herramientas Esquema menú.
4. Pulse el Run botón que aparece en la misma barra de herramientas a la izquierda del Esquema . Que construye, implementa y ejecuta la aplicación en el emulador.

Empaquetamiento Windows Phone 7 y 8

Las herramientas de línea de comandos de Windows Phone apoyan crear, construir y ejecutar nuevos proyectos. Los comandos deben ejecutarse desde un símbolo del sistema cmd o powershell.

El repo WP8 ahora incluye código para construir tanto WP7 + WP8 apps. El repositorio tiene subdirectorios para cada uno: wp7/ ywp8/.

Instalar SDK y Cordova

Descargar e instalar el SDK de Windows Phone.

Descargar y descomprimir la copia más reciente de Córdoba. Usted necesita trabajar la lib\windows-phone-8\wp7 subdirectorio, lib\windows-phone-8\wp8 contiene la versión de Windows Phone 8 de Córdoba.

Copia el CordovaWP7_x_x_x.zip archivo a la \My Documents\Visual Studio 2012\Templates\ProjectTemplates\ Directorio.

Para simplificar el desarrollo, Cordova lía un script para crear plantillas de Visual Studio. Esto permite generar rápidamente aplicaciones Cordova, y se les pueden modificar si es necesario. Los siguientes pasos muestran cómo generarlo.

Crear un proyecto

Hay 2 maneras de crear una nueva aplicación Cordova WP7 Apache o WP8.

Ejecute el archivo por lotes para crear e instalar las plantillas

La raíz de la repo contiene un createTemplates.bat archivo. Doble clic en él genera dos .zip archivos: CordovaWP7_x_x_x.zip y CordovaWP8_x_x_x.zip , donde x.x.x representa el número de versión actual. Para utilizar fácilmente estos archivos en Visual Studio, copiarlos a My Documents\Visual Studio 2012\Templates\ProjectTemplates\ .

Entonces eres capaz de crear nuevas aplicaciones Apache Cordova Windows Phone desde Visual Studio archivo → nuevo proyecto menú.

Si ejecuta el archivo por lotes desde la línea de comandos, también se puede llamar con un parámetro para instalar automáticamente

Ejecute el script:

```
>createTemplates.bat -install
```

Usar los Scripts de creación de la línea de comandos

Ejecute el create comando, especificando la ruta existente para el proyecto, el identificador de paquete estilo inversa-dominio y nombre para mostrar de la aplicación. Aquí está la sintaxis para Windows Phone 7 y 8:

```
>.\wp7\bin\create direccion del nuevo proyecto [ nombre del paquete ] [ nombre de la aplicación ]
```

```
>.\wp8\bin\create direccion del nuevo proyecto [ nombre del paquete ] [ nombre de la aplicación ]
```

Inicie Visual Studio y abra el archivo de solución (.sln) en (C:\ direccion del nuevo proyecto)

Por último se debe Construir y ejecutar

Configurar un nuevo proyecto

Abra Visual Studio Express para Windows Phone y seleccione Nuevo proyecto.

Seleccione CordovaWP7. La muestra número de versión en la descripción de la plantilla.

Dar al proyecto un nombre y seleccione OK.

Revisar la estructura del proyecto

La www características directory html , js , y css los subdirectorios y todos los demás recursos su aplicación requiere. Cualquier contenido adicional debe ser una parte del proyecto de Visual Studio, y debe insertarse como contenido.

La siguiente muestra estructura representa un 2.3.0 del proyecto, pero puede variar dependiendo de la versión instalada:

Genere el proyecto para el dispositivo

Antes de probar la aplicación en un dispositivo, el dispositivo debe estar registrado. Consulte la documentación de Microsoft para obtener más información sobre cómo implementar y probar en Windows Phone 7. Estos son los pasos básicos:

- Asegúrese de que su teléfono está conectado, y la pantalla se desbloquea.

- En Visual Studio, seleccione el dispositivo en el menú desplegable en la parte superior.
- Pulse el botón verde jugar junto al menú desplegable para iniciar depuración, o bien escriba F5.

Capítulo VI

Evaluación y trabajo a futuro.

Al concluir todo el proceso de creación de la aplicación nativa para cada sistema operativo de móviles, podemos evaluar que el funcionamiento visto en los emuladores de cada uno no presento ningún problema.

Cada prueba que se realizo fue de acuerdo al emulador, debido a que nos ahorrará tiempo y como apenas es un piloto no es necesario tener la aplicación en cada uno de los móviles, sin embargo el día del examen trataremos de presentarlo físicamente pero todo de penderá de una serie de pruebas que realizaremos unos días antes para configurar el entorno y si todo es viable se realizara.

	Compatibilidad	Fluidez en ejecución	Manejo	Almacenamiento de Datos
Android	✓	✓	✓	✓
IOS	✓	✓	✓	✓
Blackberry	✓	X	✓	✓
Windows phone	✓	X	✓	✓

Como se menciona con anterioridad los resultados son en base a emuladores y puede ser que varié ya instalado en un dispositivo.

Los parámetros presentados en la tabla anterior fueron tomados puesto que el sistema no es muy complejo y si cumple con estas funciones podemos decir que es viable para que el usuario interactúe en un 85% con el sistema.

Compatibilidad.- Se tomó en cuenta que la aplicación nativa se ejecutara en los sistemas operativos

Fluidez de Ejecución.- La rapidez con la cual se abre en diferente móvil.

Manejo.- Aquí dependerá de cada móvil de acuerdo a los recursos con los que están compuestos

Almacenamiento de Datos.- A cada uno de los casos se ingreso 300 usuarios para verificar el acceso, restricción y no se presentaron errores.

Con este proyecto esperamos ser los pioneros de esta opción que planteamos puesto que si se aprovecha bien más a delante se puede tener los siguiente:

- Incluir todas las carreras de la Fes Aragón
- Como es una web app en un futuro se pueden cargar los horarios por carrera y el estudiante los puede consultar
- Para el profesor puede ser también una herramienta en la cual pueda subir calificaciones de acuerdo al grupo y así el alumno vería por medio de la aplicación.

En fin si se aprovecha adecuadamente esta herramienta el futuro le espera es grande, claro igual se debe ir sustituyendo la tecnología ya que esta que presentamos es mas para aplicaciones pequeñas.

Conclusiones.

Una vez finalizado el proyecto podemos decir que se logró la creación de un sistema que facilite la comunicación entre los servicios que brinda el plantel y el alumno de la FES Aragón, para ellos hicimos uso de las tecnologías MVC 4 Y Phonegap, el alcance que nos brindaron nos convencieron en este tipo de proyecto, puesto que MVC nos facilitó la creación utilizando CSS, HTML y Javascript. La composición de su arquitectura de modelo visto controlador nos facilitó la identificación de errores, ya que por ejemplo podemos enfocarnos en la vista sin depender de las otras dos capas. Otra ventaja que nos presento la arquitectura es que mientras un equipo de trabajo se encargaba de la vista el otro trabajaba en la lógica del controlador; también puede concentrarse en la lógica paralelamente sin afectar en nada el desarrollo general.

La documentación encontrada al principio fue muy escasa, pero conforme avanzó el proyecto fuimos encontrando más material acerca de la tecnología.

En cuanto Phonegap podemos decir que es un Framework que al igual que MVC nos permitió tener acceso a varias plataformas porque se basa en el mismo conocimiento de CSS, HTML y JavaScript, es rápido para el desarrollo puesto que nos brinda dos maneras de empaquetar la aplicación o proyecto:

1. Es realizar una serie de procedimientos manuales que se describieron en el capítulo del desarrollo, estos procedimientos se realizan principalmente cuando uno desea mas ver el comportamiento en cada plataforma.

2.-Subir el proyecto a la herramienta llamada Phonegap build , la cual resulta ser una tarea de baja complejidad, pues solo debemos crear una cuenta, donde podemos escoger entre cuatro tipos diferentes, las cuales describiremos a continuación para que se entienda por qué elegimos la opción 1:

Cuenta como desarrollador, donde no tenemos que pagar absolutamente nada siendo completamente libre, en esta tenemos la posibilidad de tener todas las

aplicaciones públicas que necesitemos, pero solo una aplicación privada, donde solo yo tendré acceso y una aplicación colaborativa.

Cuenta Starter o de pequeñas empresas donde se paga una módica cuota de US\$12 mensuales, pero tenemos la posibilidad de pagar en el mismo tipo de cuenta todo un año por US\$120, lo que permite ahorrar US\$24. En este tipo de cuenta tenemos todas las aplicaciones publicas que sean necesarias, una privada y tres colaborativas.

Cuenta Team, donde se debe pagar una cuota de US\$30 mensuales o bien US\$300 por un año, este ultimo permite ahorrar \$US60. Con esta cuenta se tienen infinitas aplicaciones públicas, diez privadas y tres colaborativas.

Cuenta corporativa, donde la cuota a pagar son US\$90 mensuales o bien US\$900 por un año. En esta cuenta, al igual que en las otras tenemos infinitas aplicaciones públicas, veinticinco privadas y diez colaborativas.

Fue de gran utilidad porque nos permitió realizar aplicaciones relativamente sencillas como pueden ser ilustrativas e informativas y nos fue de gran utilidad para la creación de este prototipo.

La documentación encontrada del Framework principalmente fue consultada en la página oficial y ahí también pudimos consultar algunos ejemplos.

Por último podemos decir que la vida de estas tecnologías MVC 4 y Framework Phonegap será duradera debido a que lo adquirió adobe una empresa que por su gran historial en el campo de la informática habrá grandes cambios positivos y en la página oficial podemos ver varias aplicaciones creadas con Phonegap y en el caso de MVC ya podemos encontrar MVC 5 con nuevas herramientas.

Dejando a un lado ya el proyecto es necesario también comentar que la preparación que nos brindó la Universidad Nacional Autónoma de México fue la adecuada basándonos en el desarrollo del proyecto que presentamos

Sin embargo es pertinente que la educación sea menos teórica y más práctica, porque saliendo ya en el campo laboral se presentan muchas situaciones que tienes conocimiento de ello pero en la práctica es donde en ocasiones uno sufre si es la palabra adecuado para describir lo que ocurre.

Por otro lado no se le da mucho enfoque al diseño o creación de web, cuando es una rama muy importante para el ingeniero en computación.

Referencias Bibliográficas:

Herrera Ríos Emmanuel. (2012). Arrancar con HTML5. Alfaomega: México, D.F.

Bowers Michael. (2007). Pro CSS and HTML design patterns. Berkeley, California: Apress.

Jesus Bobadilla Sancho. (1999). HTML dinámico, ASP y JavaScript :a través de ejemplos. Madrid: Ra-Ma.

Morris Bruce. (1997). HTML avanzado. Madrid & México: McGraw-Hill.

Quigley Ellie. (JavaScript by example). 2011. Upper Saddle River, New Jersey & México City : Prentice Hall.

López Quijado José. (2007). Domine JavaScript. México : Alfa-Omega & Ra-ma.

Schulz Ralph G. (2009). Diseño Web con CSS. México: Alfaomega.

Orós Cabello Juan Carlos. (2011). Diseño de páginas web con XHTML, JavaScript y CSS. México, D.F. 2011