

RESUMEN

DESARROLLO DE UNA APLICACIÓN NATIVA DE ENSEÑANZA PARA IOS CON SWIFT

por

José Adair Ancona Reyes

Asesor principal: Saulo Hernández Osoria

RESUMEN DE PROYECTO DE POSGRADO

Universidad de Morelia

Facultad de Ingeniería y Tecnología

**Título: DESARROLLO DE UNA APLICACIÓN NATIVA DE ENSEÑANZA PARA IOS
CON SWIFT**

Nombre del investigador: José Adair Ancona Reyes

Asesor principal: Saulo Hernández Osoria, Maestro en Ciencias Tecnología Informática

Fecha de terminación: Abril de 2015

Problema

El presente proyecto de investigación consistió en el desarrollo de una aplicación nativa de enseñanza para el sistema operativo iOS, teniendo como objetivo principal, aumentar el conocimiento sobre el desarrollo de aplicaciones educativas para esta plataforma y sus diferentes tecnologías.

Método

Swift es un lenguaje de programación recientemente lanzado por Apple para la programación de aplicaciones en sus dispositivos móviles y de escritorio, el cuál, destaca por su rendimiento y sintaxis reducida. Durante la creación e implementación

de este proyecto, se optó por utilizar Swift para el desarrollo de una aplicación nativa, con un portal web para almacenar de manera externa el contenido de la aplicación y facilitar la administración de la información mostrada dentro de la misma.

Conclusiones

Al finalizar el proyecto, se logró implementar una aplicación para iOS y un portal web para administrar su contenido. Se adquirieron conocimientos funcionales sobre el desarrollo de aplicaciones móviles nativas para iOS, tecnologías de desarrollo web y la capacidad de utilizar estas herramientas para funcionar y trabajar en conjunto.

Universidad de Morelos
Facultad de Ingeniería y Tecnología

DESARROLLO DE UNA APLICACIÓN NATIVA
DE ENSEÑANZA PARA IOS CON SWIFT

Proyecto
presentado en cumplimiento parcial
de los requisitos para el grado de
Maestría en Ciencias Computacionales

por

José Adair Ancona Reyes

Abril de 2015


DESARROLLO DE UNA APLICACIÓN NATIVA
DE ENSEÑANZA PARA IOS CON SWIFT

Proyecto
presentado en cumplimiento parcial
de los requisitos para el grado de
Maestría en Ciencias Computacionales


por


José Adair Ancona Reyes

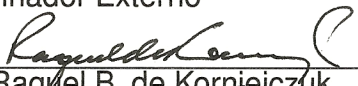
APROBADO POR LA COMISIÓN:


M.C.T.I. Saulo Hernández Osoria
Asesor principal


M.T. Carlos E. Hernández Rentería
Miembro


M.C. Daniel A. Gutiérrez Colorado
Miembro


Dr. Alonso Pérez Soltero
Examinador Externo


Dra. Raquel B. de Korniejczuk
Directora de Estudios de Posgrado

30 Abril 2015
Fecha de aprobación

DEDICATORIA

A Dios, por derramar sus bendiciones infinitas constantemente en mi vida.

A mis padres, José Ancona y Teresa Reyes, por su apoyo, cariño incondicional e invaluable enseñanza. Todo lo que soy es gracias a ellos.

A mi hermano, Jason Ancona, por su apoyo constante.

A mi novia, Mayela Gómez, por apoyarme y estar conmigo en todo momento.

A mi asesor principal, Saulo Hernández, por su amistad y apoyo para la realización de este proyecto.

A los profesores Daniel Gutiérrez y Carlos Hernández, por sus útiles consejos y amistad incondicional.

Universidad de Morelos
Facultad de Ingeniería y Tecnología

DESARROLLO DE UNA APLICACIÓN NATIVA DE
ENSEÑANZA PARA IOS CON SWIFT

Proyecto
presentado en cumplimiento parcial
de los requisitos para el grado de
Maestría en Ciencias Computacionales

por

José Adair Ancona Reyes

Abril de 2015

TABLA DE CONTENIDO

LISTA DE FIGURAS	vi
Capítulo	
I. INTRODUCCIÓN.....	1
Antecedentes	1
Antecedentes del M-Learning	1
Antecedentes del desarrollo de aplicaciones educativas	6
Definición del problema	8
Declaración del problema.....	8
Objetivos	8
Justificación.....	9
Limitaciones del proyecto	10
Delimitaciones del proyecto.....	10
Materiales, métodos y técnica	11
Definición de términos	11
Fundamento filosófico	12
II. REVISIÓN DE LA LITERATURA.....	14
El M-Learning	14
Las aplicaciones móviles educativas.....	18
Tecnologías.....	24
Mac OS	24
iOS	26
¿Qué es iOS?	26
Arquitectura de iOS.....	27
Capa Cocoa Touch	27
Capa Media.....	28
Capa Core Services	28
Capa Core OS	29
Xcode	29
Swift	31
PHP.....	33
MySQL	33

III. METODOLOGÍA.....	36
Características del proyecto	36
Planificación.....	37
Etapas del proyecto.....	37
Selección de la plataforma móvil.....	38
Arquitectura de la aplicación	42
Gestión del contenido.....	45
Presentación de la información	47
Interacción alumno-instructor	48
Implementación de las tecnologías	50
IV. CONCLUSIONES.....	60
Resumen.....	60
Conclusiones del proyecto.....	61
Resultados.....	62
V. TRABAJOS FUTUROS	63
Apéndice	
A. APLICACIÓN NATIVA IOS.....	64
B. PORTAL WEB	68
LISTA DE REFERENCIAS.....	70

LISTA DE FIGURAS

1. Capas en las que se divide iOS	27
2. Asistente de diseño de interfaces, escritura asistida y simulador de xcode	30
3. Rendimiento de swift ante otros lenguajes de programación	32
4. Diferencia de sintaxis entre Objective-c y Swift.....	32
5. Solicitud de un cliente a un servidor web con PHP	34
6. Herramienta Phpmyadmin en conjunto con MySQL para gestionar bases de datos de manera gráfica.....	35
7. Sistemas operativos móviles más usados en 2014.....	39
8. Retorno de inversión en cada plataforma móvil	39
9. Características de las diferentes arquitecturas de aplicaciones móviles.....	44
10. Características del sistema de enseñanza presencial.....	47
11. Interfaces definidas para la interacción de alumno y maestro.....	49
12. Pasos para descargar Xcode en Mac OS	52
13. Creación de la aplicación, archivos base y storyboard en Xcode.....	52
14. Pantalla principal y menú al seleccionar una clase.....	53
15. Vista principal al seleccionar la opción de video de una clase	54
16. Pantalla principal al seleccionar la opción “manual” dentro de la clase.....	54
17. Pantalla principal al elegir la sección de tarea en una clase	55
18. Vista de inicio de sesión del portal web del maestro.	56
19. Interacción entre la aplicación y el servidor web al momento de intercambiar datos.....	57

20. Función PHP que traduce la información de la base de datos a JSON.....	58
21. Archivos Swift que traducen la información de JSON para visualizarla en la aplicación y almacenarla temporalmente	59

CAPÍTULO I

INTRODUCCIÓN

Antecedentes

Antecedentes del M-Learning

Hace no mucho tiempo atrás, al hablar sobre educación y aprendizaje, era inevitable no pensar en libros, maestros y aulas de clase. Todo esto ha cambiado debido al constante uso de las tecnologías móviles como los smartphones y las tablets para realizar tareas cotidianas. Los dispositivos móviles actuales permiten realizar todo tipo de actividades tales como enviar mensajes de texto, hacer compras, leer, tomar fotografías, etc. Es por tal razón que estos dispositivos abren un nuevo panorama en la forma en la que actualmente se imparte el conocimiento, a tal grado que ahora es difícil imaginarse un ambiente en el cual estos no existan (Sarwar y Soomro, 2013).

Gracias a la reciente adopción de estas nuevas tecnologías se estima que existen cerca de 7 mil millones de suscripciones de teléfonos celulares en el mundo, representando el 95.5% de la población mundial (mobiForge, 2014).

No cabe duda que estas nuevas tecnologías han cambiado drásticamente la forma en la que las personas viven, se comunican, trabajan y socializan. Debido a esto, muchos estudiantes universitarios al llegar a los 21 años, han pasado menos de 5000 horas leyendo, han enviado y recibido alrededor de 250 000 emails, mensajes

instantáneos, de texto y más de 10 000 horas hablando por teléfono, convirtiendo al internet y a los celulares como parte integral de sus vidas. Esto en el área educativa, ha representado un conflicto entre estudiantes “nativos digitales” y maestros, catalogados como “inmigrantes digitales”, puesto que ambos tienen una idea diferente sobre cómo tiene que ser la educación.

El uso de métodos de enseñanza tradicionales con las nuevas generaciones de estudiantes será cada vez más difícil, dado que los estudiantes demandan una mayor inclusión de métodos interactivos, recursos visuales, multimedia, flexibilidad y ubicuidad en el consumo de recursos educativos (Palfrey y Gasser, 2013; Prensky, 2001).

A pesar de las ventajas que el m-learning ofrece, los profesores no aceptan del todo incluir nuevas tecnologías como herramientas aún si estas fueran totalmente accesibles y no necesitaran de ninguna ayuda técnica. Muchos de estos docentes, aunque están conscientes de las ventajas que estas tecnologías proveen para la educación, no tienen una idea de qué manera se pueden usar. Pero a pesar de las limitantes, muchos educadores reconocen la importancia de estas herramientas en sus aulas de clase, por lo que planean de manera progresiva incluir estrategias de m-learning en sus programas de enseñanza (Lang, 2009).

Uno de los primeros proyectos sobre m-learning fue el programa “From e-learning to m-learning” desarrollado por la Unión Europea en 2001, el cual consistía en desarrollar diferentes formas de llevar contenido escolar por medio de dispositivos inalámbricos como los PDAs. Se plantearon tres soluciones para poder acceder al material quedándose al final con la opción de desarrollar un portal en donde fuera

posible consumir el contenido a medida que este fuera solicitado por el usuario. Lo importante de este proyecto fue la forma en la que se desarrolló una plataforma satisfactoria de aprendizaje y su impacto en el interés por el desarrollo de futuros proyectos sobre m-learning. Aunque la tecnología de la época y el tamaño de la pantalla no permitía una buena adaptación del contenido didáctico, el programa mostró buenos resultados en la forma en la que los estudiantes interactuaban con el contenido y se relacionaban con los docentes (Landers, 2002).

En el Reino Unido durante el año 2002, debido a la gran cantidad de jóvenes de entre 16 y 20 años que se encontraban desempleados, con falta de experiencia y sin ningún interés en asistir a clases, se inició un proyecto para dar capacitación por medio de teléfonos celulares, dejando fuera otro tipo de dispositivos portátiles. Todo esto debido a que no muchas personas contaban con equipos portátiles con conexión a internet. Este proyecto fue muy importante y tuvo un gran impacto a nivel social, dado que fue uno de los primeros en preocuparse no solamente en estudiantes, sino también en el aprendizaje de la población en general (Perry, 2003).

En el año 2007 el Tecnológico de Monterrey en conjunto con la compañía de telefonía celular Iusacell, iniciaron un proyecto de aprendizaje móvil, dotando a sus estudiantes con equipos celulares BlackBerry Pearl, podcasts, archivos de audio y video, cursos en línea, conferencias, un portal para revisar sus calificaciones e intercomunicación gratuita con profesores y compañeros de clase. Aunque de igual manera el programa tuvo resultados muy positivos, los estudiantes utilizaban los dispositivos primordialmente para comunicarse con sus maestros y no como plataforma de aprendizaje (Ramos, Herrera y Ramírez, 2010).

Después del lanzamiento del iPhone, muchas empresas e instituciones se vieron interesadas en su uso de este no solo para actividades en general, sino para el sector educativo, puesto que era el primer dispositivo que integraba un navegador completamente compatible con HTML.

Todo esto fue gracias a su pantalla de gran tamaño, a la inclusión de aplicaciones nativas de terceros, de contenido en línea, de su portabilidad y fluidez que hizo del iPhone una opción verdaderamente útil para fines de aprendizaje, aunque no fue hasta la salida del iPad y su pantalla de mayor tamaño que las universidades se dieron cuenta del verdadero potencial de estos dispositivos como herramientas didácticas.

Por estas razones fue que la Universidad Cristiana de Abilene decidió proveer de iPhones e iPads a sus estudiantes, siendo la primera universidad en Estados Unidos en hacerlo. En esta iniciativa, la universidad además de usar los dispositivos para que los estudiantes pudieran recibir notificaciones sobre tareas, exámenes y cuestionarios durante las clases, también proveía dentro de otras 15 aplicaciones web diseñadas para los estudiantes, instrucciones para localizar a sus profesores, revisar sus estados de cuenta y el consumo de alimentos (Phillips, 2011).

Gracias a la disponibilidad y adopción de tecnologías móviles por gran parte de la población en general, el crecimiento del interés por actualizar y usar estrategias de m-learning por parte de las universidades e instituciones educativas ha aumentado considerablemente en los últimos años. Cada vez existen más universidades adoptando programas de aprendizaje móvil, ya sea dotando a sus estudiantes con tablets y smartphones, o exhortándolos a usar los que ya poseen.

De acuerdo con la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO por sus siglas en inglés), para el 2015 serán necesarios alrededor de 5.4 millones de docentes quienes deberán estar preparados para aprovechar al máximo las tecnologías móviles, las cuales en estas situaciones podrían ayudar a alcanzar a personas que no tienen acceso a centros educativos.

Por esta razón, es importante el uso de aparatos móviles e inteligentes para la educación, puesto que estos dispositivos, gracias al constante avance de la tecnología, hoy en día aportan tres cosas principales a la educación: Una de ellas son las características que ofrecen estos dispositivos, tales como la funcionalidad y la usabilidad. Otro es el poder cubrir las necesidades del usuario como privacidad, flexibilidad, diversión y control en el aprendizaje del contenido.

Por último se encuentran las ventajas pedagógicas que ofrecen, como la adopción de contenido móvil en las clases presenciales, también conocido como aprendizaje mezclado, el aprendizaje interactivo, la resolución de problemas y el promover la creatividad.

Las tecnologías móviles cada día están ayudando a llevar aprendizaje a más personas, minimizando el esfuerzo y los costes (concepto también conocido como efimerización). Se están volviendo más personales debido a que permiten un alto grado de interactividad entre el contenido y el usuario.

Es por esta razón, que el interés de los educadores poco a poco ha ido aumentando en pro de motivar por medio del m-learning el aprendizaje (Ally y Prieto, 2014; UNESCO, 2011).

Antecedentes del desarrollo de aplicaciones móviles educativas

Hoy en día, existen una gran variedad de tiendas y catálogos de aplicaciones móviles, la play store de google cuenta con más de 1.3 millones de aplicaciones disponibles, dejando en segundo lugar a la Apple Store con un catálogo de 1.2 millones (Statista, 2014).

Las aplicaciones móviles han dejado de ser una extensión de los sitios de escritorio, poco a poco han tomado protagonismo al punto de dejar en segundo plano el equipo en donde se usan. Las aplicaciones han extendido la funcionalidad que los smartphones y las tablets tienen por defecto, permitiendo desde poder comunicarnos por medio de redes sociales como facebook o twitter, tomar notas e incluso aprender.

Una gran cantidad de estudiantes y maestros alrededor del mundo están usando aplicaciones móviles para fines de aprendizaje, tanto es el consumo y la necesidad de aplicaciones para fines didácticos que hasta el día de hoy, el 9.95% del total de aplicaciones en la Apple Store son educativas, quedando esta categoría en tercer lugar solamente superada por las aplicaciones de finanzas, que tiene un 10.14% y por los juegos con un 21.45% del número total de aplicaciones (Filgueira Gomis, 2014; Statista, 2015).

Muchos de estos alumnos y educadores usan las aplicaciones para comunicarse, trabajar, además de su uso para propósitos didácticos. Están en constante búsqueda no solamente de aplicaciones con perfil educativo, sino de aplicaciones que realmente ayuden a mejorar el aprendizaje. Un ejemplo de esto es la aplicación para aprender idiomas Duolingo. Esta aplicación se utilizó para un experimento de 8 semanas con el fin de enseñar español a un grupo de personas de 18 años, residen-

tes de Estados Unidos. Entre los requisitos de esta investigación, se encontraba el que los participantes no fueran de origen hispano y que tuvieran como lengua materna el inglés. Los participantes hicieron un examen para evaluar su conocimiento al principio y al final del periodo de 8 semanas. El resultado final del experimento fue que al estudiar por lo menos dos veces por semana con Duolingo, los participantes mostraron un resultado de 91.4 puntos de aumento en su aprendizaje del español, esto debido a que la aplicación utiliza recursos que los usuarios encuentran motivantes y atractivos (Vesselinov y Grego, 2012).

Otro ejemplo es la aplicación Lumosity, considerada una de las mejores aplicaciones para ejercitar el cerebro. Lumosity se destaca debido a su gran variedad de juegos basados en ejercicios para mejorar diferentes áreas cognitivas tales como la memoria, la resolución de problemas, tiempo de respuesta y atención, Una de las cosas que hace de Lumosity una opción relevante, es que detrás de su desarrollo se encuentran expertos en neurociencia, además que se ha utilizado como herramienta de estudio en diferentes escenarios dando resultados significativos (Hardy, Farzin, y Scanlon, 2013).

Las aplicaciones móviles educativas cada vez están más al alcance de todos. Hoy en día existen un gran número de entidades interesadas en promover conocimiento, dedicadas al desarrollo y adaptación de contenido didáctico para móviles tales como TED, Lynda, Udemy, Coursera y Udacity, solo por mencionar algunas. Estas plataformas facilitan en gran manera el acceso a diferentes cursos de manera gratuita o de pago, ofreciendo una gran variedad de temas, foros de discusión, referencias y herramientas que se adaptan a las necesidades del usuario para su como-

didad y motivación. Teniendo como fin principal, mejorar la experiencia de aprendizaje del usuario y la disponibilidad del contenido de enseñanza.

Definición del problema

La constante búsqueda de información, el uso de internet, el surgimiento de nuevas tecnologías, la necesidad de inmediatez al obtener información, el catálogo extenso de fuentes variadas de conocimiento, la portabilidad, la usabilidad y la accesibilidad, han hecho que el uso de dispositivos móviles como los smartphones y las tablets jueguen un rol muy importante en el aprendizaje en general. Por lo tanto, es importante que existan aplicaciones móviles de enseñanza. Esto con el tiempo ha ido mejorando y aunque existen plataformas móviles de aprendizaje para diferentes temas, aún persiste la necesidad de una aplicación dedicada a la enseñanza de conceptos sobre redes e internet.

Declaración del problema

Teniendo como objetivo el aprendizaje del desarrollo de aplicaciones móviles para enseñanza sobre conceptos básicos sobre redes, seguridad, internet y direccionamiento IP, el presente proyecto plantea hacer uso del lenguaje de programación Swift para diseñar una aplicación nativa para la plataforma iOS con un portal web para administrar su contenido.

Objetivos

Los objetivos del proyecto son los siguientes:

1. Explorar el campo del desarrollo de aplicaciones móviles educativas, sus diferentes características, posibilidades e implementación.

2. Aprender y profundizar sobre el proceso de programación de aplicaciones nativas para iOS por medio del nuevo lenguaje de programación desarrollado por Apple llamado Swift.

3. Desarrollar una aplicación nativa para iOS y un portal web para administrar el contenido de la aplicación.

Justificación

En este proyecto, se propone el desarrollo de una aplicación móvil para la enseñanza de conceptos de redes e internet desarrollada para iOS. Toda la información podrá ser administrada remotamente desde un portal web por el o los maestros encargados del material educativo. En la aplicación, el curso se encontrará dividido en diferentes clases o capítulos en video para su mejor comprensión, que pueden ser visualizados teniendo conexión a internet o ser descargados al dispositivo dentro de la misma aplicación para ser visualizados sin conexión.

El curso tiene el propósito de comenzar con temas y conceptos básicos para que de esta forma el usuario pueda comprender de manera concreta el contenido de cada tema. El usuario tendrá completa libertad de escoger el horario o las clases que sean de mayor interés. Después de terminar una o todas las clases del tema seleccionado, se pretende incluir en la misma sección un manual o libro en formato PDF como complemento a la clase o apuntes extras. Además de incluir un área en donde el responsable de impartir las clases pueda agregar una tarea en formato PDF para que el estudiante pueda evaluar el conocimiento obtenido al finalizar la clase. El documento podría consistir de texto, imágenes o links a sitios web.

Los dispositivos móviles y sus aplicaciones han demostrado tener el potencial

necesario para llevar a cabo diferentes funciones, siendo la principal el hacer que el contenido este disponible al momento en el que el usuario lo necesite, aprovechando de manera óptima el tiempo, ya sea dentro de un aula de clase o fuera de ella.

Limitaciones del proyecto

Las limitaciones del proyecto fueron las siguientes:

1. Swift es un lenguaje de programación recientemente lanzado al público, por lo que existe muy poca documentación y recursos para poder profundizar de mejor manera en aspectos más avanzados de la programación de aplicaciones para iOS usando este lenguaje.

2. Debido a la falta de una licencia de desarrollador que ofrece Apple, la cual es necesaria para probar la aplicación en un dispositivo, la aplicación solo podrá ser ejecutada en un simulador y no será subida a la tienda de aplicaciones.

Delimitaciones del proyecto

Dado que el proyecto además de crear una aplicación para aprender conceptos sobre redes e internet, tiene la intención de aumentar el conocimiento sobre el desarrollo de aplicaciones de enseñanza para iOS, las delimitaciones del estudio fueron las siguientes:

1. Debido a que existe una gran variedad de lenguajes y plataformas para el desarrollo de aplicaciones nativas, el lenguaje seleccionado para el desarrollo de esta aplicación es Swift por ser un lenguaje de programación recientemente lanzado y la plataforma iOS.

2. Tomando en cuenta la escasez de documentación, se presentará una apli-

cación con algunos temas de ejemplo y funciones elementales planteadas, por lo que sólo se desarrollará un prototipo funcional.

Materiales, métodos y técnica

Los materiales con los que se trabajó en este proyecto fueron una computadora portátil con Mac OS X 10.10.2 Yosemite instalado; Xcode 6.1, el cual contiene todos los módulos, simuladores y referencias necesarias para desarrollar aplicaciones nativas para iOS; Se utilizará el sistema operativo iOS en su versión 8.2; El lenguaje de programación Swift, lenguaje para desarrollar aplicaciones para iOS y Mac OS X.

Los métodos y técnica utilizados fueron las siguientes: (a) estudio de tecnologías, (b) desarrollo de una aplicación de iOS para el aprendizaje de conceptos de redes (c) desarrollo de un portal web y (c) prueba concepto.

Definición de términos

A continuación se definen los conceptos que este documento presenta como parte de la terminología que describe el proyecto:

1. M-Learning (Mobile Learning): Forma de educación a distancia que utiliza como plataforma principal un dispositivo móvil.

2. Aplicación móvil (App): Aplicación de software para dispositivos móviles que permiten al usuario realizar actividades con fines productivos o de entretenimiento.

3. Smartphone: Teléfono celular con funciones de una computadora tales como contener y reproducir contenido multimedia, acceso y redacción de mensajes instantáneos, correo electrónico, descarga de aplicaciones, grabar audio y tomar fo-

tografías.

4. Tablet: Dispositivo con capacidades y características similares a un smartphone con la diferencia de tener un tamaño mayor a éste y ser más pequeña que una computadora.

5. Dispositivo móvil: Cualquier dispositivo electrónico que permita al usuario realizar sus actividades y conectarse a internet de manera inalámbrica.

6. Framework: Grupo de conceptos, practicas o herramientas que trabajan en conjunto para resolver un problema.

7. Mac OS X: Sistema operativo basado en Unix desarrollado por Apple para sus computadoras portátiles y de escritorio.

8. iOS: Sistema operativo de la compañía Apple para todos sus dispositivos móviles como el iPad, el iPhone y el iPod.

9. Xcode: Entorno de desarrollo integrado para el diseño de aplicaciones nativas para dispositivos móviles con sistema operativo iOS.

10. Swift: Nuevo lenguaje de programación lanzado por Apple enfocado principalmente en el desarrollo de aplicaciones nativas para dispositivos con sistema operativo iOS o Mac OS X.

Fundamento filosófico

Dios dotó al hombre de una sofisticada forma de comunicarse con su creador, pero a causa del pecado el hombre rompió esta comunicación. Dios en su infinito amor, le recuerda al hombre en Jeremías 29:12 que puede comunicarse con Él por medio de la oración. Tomando en cuenta lo anterior, el ser humano tiene la obligación de transmitir el mensaje tan importante de amor que el Señor nos ha enseñado.

Por esta razón, es necesario hacer uso de una gran herramienta como lo es la tecnología para transmitir el mensaje de Dios a todas partes, haciendo uso de todo lo que provee y de esta manera, recordar a las personas que Dios quiere comunicarse con ellos y suplir sus necesidades.

CAPÍTULO II

REVISIÓN DE LA LITERATURA

En este capítulo se presentan los conceptos principales de este proyecto y el funcionamiento básico detrás de las tecnologías necesarias para su elaboración. Se mencionan las ideas y propósitos principales del m-learning, el desarrollo de aplicaciones educativas, los dispositivos móviles de Apple, su sistema operativo Mac OS, el lenguaje de programación Swift y la herramienta Xcode para el desarrollo de aplicaciones para iOS y Mac OS X.

El M-Learning

El aprendizaje móvil o “m-learning” se describe como una forma de educación a distancia que utiliza como plataforma principal un dispositivo móvil, el cual permite que el usuario no se vea en la necesidad de asistir a un aula de clase o a un lugar en específico, teniendo como ventaja que el proceso de aprendizaje pueda llevarse a cabo en cualquier lugar a cualquier hora.

Aunque la creciente adopción de dispositivos móviles en los últimos años hace parecer que el concepto de m-learning es reciente, la realidad es que no es algo nuevo. La idea fue desarrollada a finales de los años 60's por Alan Kay, quien desarrolló un concepto conocido como dynabook. Kay visualizó un dispositivo especialmente diseñado para estudiantes que similar a un libro, fuera capaz de desplegar imágenes y textos. Aunque este concepto nunca fue desarrollado, es fácil notar que

ha servido como base para el diseño de dispositivos móviles modernos, principalmente las tablets. Después del concepto del dynabook, las computadoras se han hecho más personales, accesibles y eficientes. En años recientes, la tecnología ha avanzado de tal forma que se han desarrollado dispositivos móviles, los cuales actualmente permiten al usuario realizar tareas o actividades de aprendizaje sin depender de un lugar o un horario en específico, de esta forma, el aprendizaje puede tomar lugar en cualquier parte y en cualquier momento (Pachler, Bachmair, y Cook, 2009).

El m-learning suele entenderse o ser visto mayormente como aprendizaje electrónico o e-learning, y aunque existe relación entre estos dos conceptos, ambos tienen características propias que los diferencian uno del otro. Una de las principales diferencias es el tiempo y espacio en el que se espera que el aprendizaje tome lugar y la duración del consumo del contenido didáctico. La mayoría del material de e-learning disponible está diseñado a manera de sesiones con límite de tiempo que pueden ir desde 15 minutos hasta 2 horas. Esto se debe a que el aprendizaje toma lugar en una computadora de escritorio o portátil en algún lugar en específico, a diferencia del m-learning que está diseñado para que el material sea estudiado en razón de las necesidades del estudiante.

Otra diferencia importante son las características únicas e importantes que proveen los dispositivos móviles tales como la geolocalización, los acelerómetros, cámaras y otras funciones que convierten a estos aparatos en herramientas multifuncionales, pero lo más importante, es la posibilidad que existe para desarrollar programas que utilicen estas funciones para mejorar la interacción con el estudiante. Aunque ambos conceptos tienen diferencias y similitudes no necesariamente quiere

decir que uno es mejor que otro. Cada uno cumple una función específica en diferentes áreas y ayudan a que los estudiantes y diferentes entidades cuenten con diferentes opciones que puedan suplir sus necesidades (McQuiggan, McQuiggan, Sabourin y Kosturko, 2015; Udell, 2014).

El concepto de m-learning tiene tres fases esenciales que son: (a) el enfoque a los dispositivos móviles, (b) el aprendizaje fuera del salón de clase y (c) dar flexibilidad al aprendizaje del estudiante. El enfoque a los dispositivos móviles se caracteriza por la búsqueda de aparatos inalámbricos que puedan ser usados como herramientas en un contexto educativo, que ayuden a recibir o mandar resultados relacionados con el aprendizaje o capacitación en diferentes áreas. El aprendizaje fuera del salón de clase puede ser relacionado con viajes escolares y visitas a museos. Tomando en cuenta la primera fase, los dispositivos móviles ofrecen diferentes opciones y contenidos para continuar el aprendizaje fuera del aula por medio de aplicaciones, elementos multimedia, documentos y demás recursos didácticos variados. Por último, la flexibilidad de aprendizaje consiste en permitir que el usuario tenga la libertad de gestionar el consumo de contenido didáctico, por medio del diseño de ambientes móviles ideales para éste fin y el desarrollo de contenido relevante para el estudiante (Udell, 2014).

Poco a poco la sociedad ha ido cambiando y se encuentra constantemente en movimiento, de ésta misma forma también se encuentra conectada a sus amigos, familiares, lugares de trabajo y fuentes de información. Para muchas personas es importante tener acceso a dispositivos que les permitan realizar actividades desde cualquier lugar tales como agentes de ventas, asesores, transportistas y ejecutivos

de alto rango los cuales, es de vital importancia poder comunicarse en todo momento ya que es parte no solamente de su trabajo, sino del estilo de vida que el ambiente que les rodea les exige. Gracias al constante avance de la tecnología las personas cada vez están más equipadas para poder realizar este tipo de actividades sin ningún problema y esto afecta no solamente el tiempo, sino el lugar donde el aprendizaje se lleva a cabo. La necesidad de un aprendizaje más rápido es evidente debido a que la tecnología cambia rápidamente, así mismo los procedimientos, estrategias y tecnologías adoptadas por las escuelas y las empresas. El tiempo disponible poco a poco se ha ido reduciendo por la necesidad que existe de estudiantes y trabajadores que realicen más de una sola tarea a la vez.

Los dispositivos móviles cada vez se han vuelto más asequibles. Debido a esto, muchas personas cuentan con este tipo de dispositivos, por lo que ya se cuenta con el ambiente necesario para poder implementar un aprendizaje móvil. Tomando en cuenta lo anterior, ya no es necesario que las instituciones o empresas provean de estas herramientas ya que actualmente, muchos estudiantes y trabajadores usan las tecnologías móviles en su rutina diaria. La globalización es un factor importante en esto, puesto que el uso de estos dispositivos y sus características en conjunto con las redes sociales permiten que las personas puedan comunicarse prácticamente de manera inmediata, por lo que proveen no solamente un ambiente ideal para implementar el m-learning sino también eficiencia, portabilidad e inmediatez en el consumo de contenido educativo (Berge y Muilenburg, 2013; Udell, 2014).

El m-learning ha provocado muchos cambios, los principales y más importantes son: (a) más disponibilidad del contenido, (b) reducción del tiempo de aprendiza-

je, (c) mejorar el contexto y ambiente del aprendiz, (d) reducir la cantidad de información, (e) concentrar la información importante, (f) mejorar la experiencia y la calidad del aprendizaje y (g) dar control al usuario del contenido. De esta forma el m-learning permite no solamente dar al estudiante la libertad y facilidad de aprender, sino también llegar a más personas de manera fácil, rápida y más eficiente.

El aprendizaje comienza en cualquier lugar en donde alguien formule preguntas y tenga la intención de buscar respuestas. A manera que la información se vuelve más accesible desde los dispositivos móviles, el aprendizaje cada vez más va tomando lugar fuera de un aula de clase, ya sea durante una conversación casual, un viaje, una salida escolar o al momento de realizar alguna actividad de entretenimiento como ver una película o leer un libro. El m-learning ha permitido que el aprendizaje pueda gozar de nuevos recursos didácticos más interactivos y actualizados, también ha incrementado la participación por parte de los alumnos y ha hecho que la experiencia de aprender sea más personal fuera del aula. Poco a poco los docentes están mezclando las nuevas tecnologías con los métodos de enseñanza actuales, logrando que los estudiantes puedan aprender de manera efectiva en su día a día (Udell, 2014).

Las aplicaciones móviles educativas

Una aplicación móvil también abreviada “app”, puede definirse como un software específicamente diseñado para correr en un dispositivo portátil como un smartphone o una tablet. Estas aplicaciones al momento de ser descargadas por el usuario, se ejecutan nativamente en el sistema operativo de los diferentes tipos de dispositivos, aprovechando de una mejor forma las diferentes características que es-

tos contienen. Tomando en cuenta lo anterior, una aplicación móvil educativa es cualquier aplicación que tenga como intención principal promover el aprendizaje (Salz y Moranz, 2013).

La necesidad de flexibilidad en la forma en que se comparte la información cada vez es mayor, y con la evolución de los clásicos dispositivos de escritorio o dispositivos portátiles personales a celulares inteligentes y tablets, han influido para que todo el contenido ya sea de entretenimiento o educativo que actualmente se encuentra disponible, esté adaptado para su visualización en un dispositivo móvil. Cada vez existen más usuarios que se conectan solamente por estos dispositivos, por esta razón las interfaces móviles se han vuelto una opción obligatoria para los que producen contenido en general, ya que les permite llegar de manera más puntual a sus usuarios, resumiendo la información pero sin dejar a un lado lo que el usuario necesita (Kim, 2013).

Las primeras aplicaciones móviles fueron de la mano con los primeros teléfonos inteligentes y eran desarrolladas solamente por los fabricantes de estos equipos. La presión y la competencia sobre quien diseñaba mejores teléfonos se encontraba muy reñida, por lo que tomando esto en cuenta, los creadores de teléfonos decidieron no solamente desarrollar el hardware sino también desarrollar y controlar el software, esto hizo que no existiera la posibilidad de incluir programas o aplicaciones desarrolladas por terceros. En 1983 Motorola lanzó al mercado el DynaTAC, considerado como el primer celular, el cuál incluía una pequeña aplicación para guardar contactos en su sistema operativo.

En un principio las aplicaciones como el calendario, algunos juegos programa-

dos en java y una calculadora eran considerados como aplicaciones móviles. Durante este periodo Nokia vio la oportunidad de incluir versiones sencillas de juegos de los años 70's como Pong, Snake y Tetris en sus teléfonos haciendo que se volvieran muy populares entre los consumidores. Estos teléfonos cambiaron radicalmente la forma en la que las personas pensaban sobre el poder comunicarse, ya que a medida que los precios de estos equipos iban descendiendo, la autonomía de estos junto con las áreas de recepción fueron mejorando y su uso entre los consumidores fueron ganando terreno. Pronto los celulares dejaron de ser un lujo y pasaron a ser una necesidad (Clark, 2012).

Al aumentar la popularidad de los celulares entre los usuarios de la época, estos empezaron a presionar a las compañías a proveer una mayor cantidad juegos y funciones de alta calidad, pero las compañías no tenían los recursos necesarios para crear todas las aplicaciones que los consumidores exigían, por lo que necesitaban una forma de llevar contenido a los equipos con las herramientas que ya tenían disponibles. Aunque se pensó usar el internet como canal para difundir contenido a los aparatos de las personas, los teléfonos no contaban con la funcionalidad, capacidad de almacenamiento o la autonomía suficiente para poder mostrar el contenido debidamente, por lo que decidieron usar tecnología WAP, una versión más ligera de HTTP el cuál es uno de los protocolos más básicos de la web. Esta solución fue suficiente por un tiempo para llenar la necesidad de los clientes puesto que permitía difundir contenido desarrollado por terceros por medio de canales oficiales de las compañías, facilitando de esta forma el regular y llevar de una mejor manera contenido a los usuarios.

Durante este periodo de tiempo las aplicaciones más populares fueron canales que difundían tonos de llamada y fondos de pantalla, permitiendo a los usuarios por primera vez poder personalizar sus teléfonos.

Aunque WAP cumplía con su trabajo, los desarrolladores no podían llevar a los usuarios contenido de calidad debido a las grandes limitaciones de esta tecnología y aunque los canales de sitios como la CNN o ESPN fueron muy populares y exitosos, era muy difícil poder crear aplicaciones, ya que no existía una guía como tal que explicara cómo hacerlo. WAP era muy lento y frustrante. Para los usuarios escribir direcciones de internet muy largas en un teclado numérico era muy cansado. Muchas de las aplicaciones no estaban optimizadas para soportar diferentes tipos de celulares, por lo que el usuario debía llenar un formulario por medio de su dispositivo solicitando una versión específica para su modelo de equipo, tamaño y funciones (Clark, 2012; Conder y Darcey, 2010).

A medida que las memorias se volvieron más baratas y las baterías mejores, fue así como los PDA's y otro tipo de dispositivos portátiles que podían correr versiones compactas de windows y linux se volvieron más populares, ya que eran sistemas con las que las personas estaban más familiarizadas. Sin embargo, uno de los primeros equipos que es considerado como el primer smartphone fue presentado para uso general en 1993 por la empresa RIM, el cual estaba cargado con características como un calendario, una calculadora, un reloj mundial y una agenda de contactos.

Pero no fue hasta que en el año 2002 RIM presentaría uno de los más grandes avances en el desarrollo de aplicaciones móviles, el teléfono inteligente BlackBerry el cual venía integrado con innovador concepto de una aplicación de email

inalámbrico (Clark, 2012).

Poco a poco las compañías se dieron cuenta que para poder seguir vendiendo celulares, debían dar más participación al desarrollo de software por parte de personas ajenas a ellas, por lo que esto permitió que los desarrolladores se interesaran en la programación de contenido para estos equipos. Pronto, empezaron a surgir nuevas plataformas con una gran cantidad de aplicaciones móviles como el sistema operativo PALM, siendo uno de los primeros acompañado por RIM BlackBerry OS y Symbian desarrollado por Nokia, Sony Ericsson, Motorola y Samsung (Pelletier, 2013).

Antes de la introducción del iPhone en el año 2007 aunque el entorno para desarrollar aplicaciones de terceros ya estaba muy avanzado, seguía estando muy controlado, puesto que cada teléfono contaba con su propia estructura de programación y la interacción entre los aparatos y las aplicaciones era extremadamente limitada. Todo esto cambió con el lanzamiento del iPhone, puesto que era el primer smartphone que incluía un navegador que permitía visualizar una página web tal como se veía en una computadora de escritorio. Apple en un principio, animó a los desarrolladores a crear aplicaciones web para el iPhone, al inicio esto fue suficiente, pero los desarrolladores no estaban satisfechos porque al ser aplicaciones web, no era posible acceder a todas las características del teléfono.

Fue entonces en el año 2008 que Apple anunció que distribuiría aplicaciones nativas de terceros para el iPhone y un SDK (Kit de desarrollo de software). También se dieron a la tarea de crear una plataforma de desarrollo de aplicaciones llamada Xcode y de igual manera, crearon lo que sería el principio de las tiendas de aplica-

ciones, la Apple Store (Bucanek, 2013).

Después de esto, más y más tiendas de aplicaciones y sistemas operativos han sido lanzadas para el consumo general. Los cambios constantes en la tecnología han permitido que los dispositivos móviles vayan ganando más funciones, mejor autonomía, pantallas de mayor tamaño, conexión a internet y mayor apoyo por parte de las compañías a los desarrolladores de aplicaciones, creando una constante lucha por sacar el mejor provecho de estos equipos. El interés no solo de las personas adultas, sino de gente de todas las edades por el uso de estos aparatos y sus aplicaciones ha resultado en que constantemente se cree contenido personalizado para el usuario (Salz y Moranz, 2013).

Gracias a la gran interacción que las aplicaciones y los aparatos móviles permiten experimentar a los usuarios, ha permitido su uso como herramientas en la educación. Las aplicaciones proveen velocidad, seguridad y simplicidad. Están diseñadas para iniciar casi de manera inmediata e iniciar remotamente desde cualquier dispositivo sin importar las exigencias del usuario. Son seguras, pues corren en un dispositivo diseñado para uso personal y aunque también se pueden usar en aparatos compartidos, la información queda segura dentro del aparato. Permiten llevar un control más personal del contenido, ya sea que se incluya una función para compartir los logros alcanzados, o por medio de alarmas o notificaciones locales o remotas enviadas al dispositivo por medio de la aplicación, logrando de esta manera, mantener un gran nivel de interés por parte del estudiante. Por último, hacen de lo difícil algo más sencillo, interesante, limpio y más eficiente (McQuiggan et al., 2015).

Las aplicaciones móviles ofrecen la posibilidad de atraer y fomentar el debate

y resolución de problemas por medio de foros de chat y de discusión, sin mencionar la posibilidad de consultar información y resolver dudas sin la necesidad de ir al salón de clase. Los maestros pueden formar parte de estas discusiones y poder guiar a los estudiantes cuando sea necesario. Estas aplicaciones pueden ayudar a mostrar, organizar e incluso diseñar la forma en la que se imparte la información para que se pueda adaptar al estudiante (Udell, 2014).

Muchas compañías teniendo esto en cuenta, están en constante desarrollo de aplicaciones para teléfonos inteligentes, computadoras y tablets con fines de aprendizaje y enseñanza, utilizando recursos como cuestionarios, ejercicios en línea, libros electrónicos, videos y contenidos interactivos que ayudan a estimular el aprendizaje en una forma en la que los libros no pueden. Aunque las calculadoras, los proyectores y las computadoras eran vistas como formas revolucionarias de enseñanza en el aula de clase. Actualmente los celulares, las tablets, las aplicaciones móviles y las redes sociales, han pasado de ser una simple moda a convertirse en materiales efectivos para el aprendizaje en las escuelas ayudando de esta forma, a que alumnos y maestros puedan tener una mejor experiencia dentro y fuera del aula de clase, permitiendo que existan un impacto positivo en la calidad de enseñanza y de aprendizaje (Zammetti, 2013).

Tecnologías

Mac OS

Mac OS (Macintosh Operating System) es el nombre que lleva el sistema operativo creado por Apple para su línea de computadoras portátiles y de escritorio. En sus inicios en 1985, Apple diseñó completamente la arquitectura de su sistema inclu-

yendo menús y ventanas a diferencia de los primeros sistemas operativos que solamente eran líneas de comandos y no fue hasta el 2001 que Apple decidió renovar por completo su sistema operativo lanzando el OS X y cambiando su arquitectura completamente a Unix. Mac OS aunque es un sistema operativo bastante funcional y robusto, representa la unión de diferentes tecnologías de otros sistemas operativos. En el núcleo de éste se encuentra Unix, que fue desarrollado por AT&T alrededor de 1970, el cuál hasta el día de hoy es usado en computadoras funcionando como servidores de red o máquinas de uso general. Durante la evolución de Unix, éste se ha dividido en varias distribuciones, todos dirigidos a diferentes grupos de personas con múltiples necesidades y objetivos. La distribución que forma parte esencial de este sistema se conoce como Darwin y a pesar de que Apple hace el intento de esconder esto de la mayoría de los usuarios, es posible ejecutar instrucciones de Unix en la terminal de comandos del sistema (Warner y Privat, 2011).

Una de las principales características de OS X es que se encuentra específicamente diseñado por Apple para que solamente pueda ser instalado en computadoras Mac, argumentando que de esta forma se asegura el correcto funcionamiento del sistema. Después de que Apple decidiera cambiar los procesadores de sus computadoras de PowerPC a Intel, el proceso para poder instalar este sistema operativo en otras computadoras con procesadores de la misma marca cada vez ha resultado más viable, aunque con sus limitantes debido a las diferencias de hardware. Hoy en día, existe una gran multitud de manuales y servicios que permiten la instalación o el uso de este sistema sin la necesidad obligada de adquirir hardware de Apple para lograrlo, existe un término conocido como “Hackintosh”, el cual consiste en la instala-

ción del sistema operativo de Apple en un hardware no diseñado por ellos. También existen sitios web como MacinCloud que ofrecen la renta de una máquina virtual con diferentes versiones de Mac OS para su uso en línea (Sadun y Wardwell, 2014).

Aunque existen diferentes maneras de poder usar este sistema operativo sin hardware oficial, Mac OS X se vuelve un requisito importante al momento de querer desarrollar aplicaciones para iOS ya que como requisitos principales se encuentran (a) una computadora Mac con un procesador Intel, (b) Mac OS X 10.9 o superior (Yosemite) y la versión más actualizada de Xcode (Bakir, 2014).

iOS

¿Qué es iOS?

Antes conocido como iPhone OS, es el sistema operativo que Apple usa para todos sus dispositivos móviles derivado de Mac OS por lo que está basado en Unix. iOS corre actualmente en el iPhone, iPad, iPod Touch y Apple TV. Este sistema operativo, se destaca porque hizo que la interacción con el primer iPhone fuera por medio de gestos como deslizar y tocar. Aunque esta tecnología no era del todo nueva, Apple popularizó este tipo de interacción con los dispositivos móviles. iOS viene preinstalado en los dispositivos de esta compañía con una gran variedad de aplicaciones tales como mapas, un navegador, una tienda de aplicaciones, un gestor de emails, un reproductor de música, llamar y mandar mensajes (Bucanek, 2013).

El kit de desarrollo de iOS (SDK) contiene todas las herramientas y las interfaces que se necesitan para desarrollar, instalar, ejecutar y probar aplicaciones nativas en un dispositivo con iOS. Este tipo de aplicaciones se desarrollan usando frameworks y el lenguaje de programación Objective-C o Swift. A diferencia de las web

apps, las aplicaciones nativas están instaladas físicamente en el dispositivo y por lo tanto, están siempre disponibles para el usuario (Apple, 2014).

Arquitectura de iOS

iOS se puede tomar como un intermediario entre el equipo físico y las aplicaciones que se programan. Como en otros dispositivos, las aplicaciones no pueden interactuar directamente con el hardware del dispositivo, sino que se comunican a través de un conjunto de interfaces del sistema bien definidas. Si vemos la Figura 1, la arquitectura de iOS puede ser vista como en capas en donde las capas de nivel más bajo contienen servicios importantes para el funcionamiento y las de más alto nivel proveen a los niveles más bajos de tecnologías y servicios más sofisticados.

Capa Cocoa Touch

Cocoa Touch es una capa de iOS que contiene frameworks para definir la interfaz de usuario y que provee Apple para el desarrollo de aplicaciones para iOS.

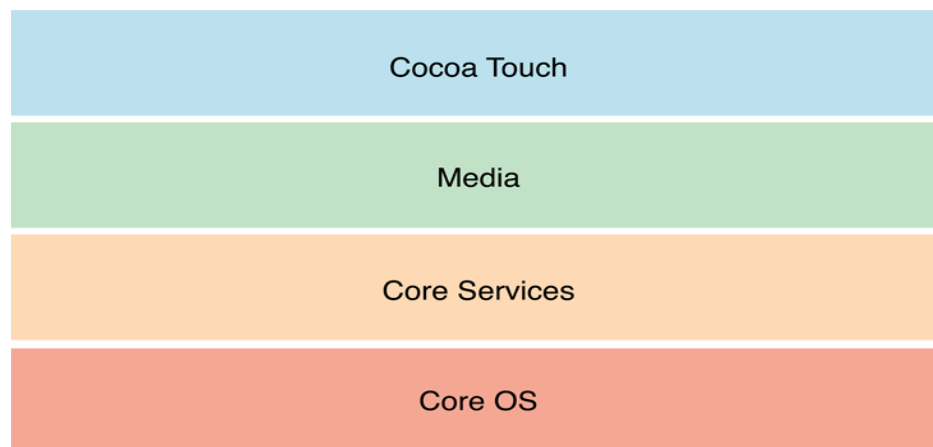


Figura 1. Capas en las que se divide iOS.

Está escrita mayormente en Objective C basándose en Mac OS. El conjunto de frameworks incluido en esta capa, definen la apariencia de las aplicaciones y cosas como el multitasking, notificaciones, gestos, etc. Además de las funciones ya mencionadas, incluye frameworks como Auto Layout, que facilitan el cambio de dimensiones de imágenes y texto para diferentes resoluciones de pantalla. Storyboards, el cual tiene la función de permitir diseñar de manera visual la interface al momento de desarrollar una aplicación. AirDrop, que permite compartir documentos, imágenes, url's y otros diferentes tipos de archivos con otros dispositivos. Estos y otros frameworks en su mayoría de carácter visual vienen incluidos en esta capa.

Capa Media

En esta capa se incluyen todas las herramientas que tienen que ver con la gestión de gráficos, audio y video de las aplicaciones, para que puedan sonar y verse sin ningún problema. Dentro de esta capa se encuentran frameworks como UIKit el cual es el responsable en dar soporte al trazo de imágenes y animar el contenido vistoso de las aplicaciones. Core Image, que provee la manipulación de video e imágenes. Metal, que permite una interacción más cercana con el GPU del dispositivo móvil, dando como resultado mayor rendimiento en la ejecución de aplicaciones de gran tamaño.

Capa Core Services

Esta capa es la que está a cargo de controlar los servicios fundamentales del sistema que las aplicaciones nativas usan. La capa Cocoa Touch se apoya en gran manera en esta capa para poder funcionar. En esta capa se encuentran frameworks

como Foundation y Foundation Core, que como su nombre lo dice es la base en la que descansan muchas aplicaciones. Entre algunas de las funciones de este framework se encuentra el hacer más fácil compartir entre aplicaciones datos y código, permitiendo de esta forma evitar la duplicación de información y obtener un mejor desempeño. iCloud Storage, que permite guardar contenido, datos de aplicaciones, documentos y otro tipo de archivos en iCloud. In-App purchase, que provee la funcionalidad de hacer transacciones monetarias por medio de contenido extra disponible dentro o extra de la aplicación, que al concretarse la transacción es descargado como complemento o nueva funcionalidad a esta.

Capa Core OS

En esta capa se encuentran las características de bajo nivel en la que la mayoría de las otras tecnologías están basadas. Entre las principales funciones de los frameworks que tiene, se encuentra el poder usar el acelerómetro, el bluetooth, encriptación de datos, soporte para 64-bits, entre otras cosas (Apple, 2014).

Xcode

Es una suite de herramientas creada por Apple para el desarrollo de aplicaciones para Mac OS y iOS. Se puede descargar gratuitamente de la tienda de aplicaciones integrada en el mismo sistema operativo Mac. La interface principal de Xcode además de un editor de código, incluye un ambiente integrado de desarrollo (IDE). También incluye una gran cantidad de la documentación oficial de Apple y un editor visual el cual permite mostrar varias vistas de un archivo o diferentes archivos relacionados dentro de la misma ventana. Como se puede observar en la Figura 2, Xco-

de dentro de su menú integrado incluye entre otras herramientas, tres que ayudan en gran manera a mejorar la escritura de aplicaciones: (a) escritura de código asistida, (b) asistente de diseño de interfaces y (c) simulador (Kelly y Nozzi, 2013).

La escritura de código asistida que viene incluida en Xcode, revisa el código mientras éste es escrito al momento de estar editando código, ya sea Objective C, Swift, C, C++ o una mezcla. Cuando se encuentra un error, el editor de código destaca el código erróneo y en ciertas ocasiones ofrece una solución.

El asistente de diseño de interfaces usa un asistente para crear los elementos visuales de las aplicaciones por medio de ventanas, botones, controles, menús, y otros elementos dentro de una biblioteca de objetos programables. También incluye Storyboards, los cuales sirven para definir las acciones y las transiciones entre escenas por medio de conectores visuales.

El simulador viene con todo lo necesario para poder probar los aspectos básicos de una aplicación.

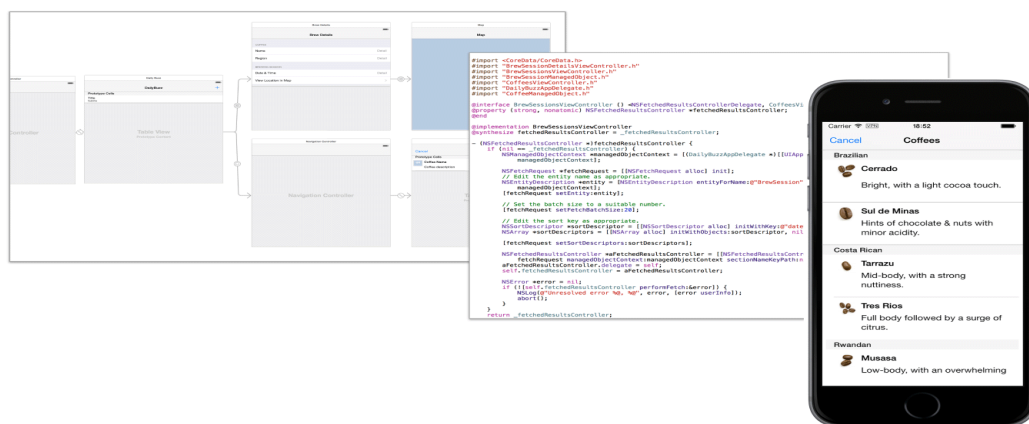


Figura 2. Asistente de diseño de interfaces, escritura asistida y simulador de Xcode.

A excepción de las notificaciones push, el acelerómetro, el gps entre otras cosas, casi todas las funciones básicas de una aplicación pueden ser probadas desde el simulador de Xcode. Además, incluye un simulador para diferentes versiones de iOS con sus diferentes dispositivos correspondientes (Varma, 2015).

Swift

Es el nuevo lenguaje de programación de aplicaciones para iOS y OS X que se basa en lo mejor de C y Objective-C, sin las limitaciones de la compatibilidad de C. Swift está diseñado no para reemplazar el código ya escrito en Objective-C, sino para complementarlo ya que es posible trabajar en conjunto con ambos.

Como se puede observar en la Figura 3, Swift tiene mucho mejor rendimiento que Objective-C, esto se debe a que Objective-C toma más tiempo en compilar a diferencia de Swift que es un lenguaje más cercano al hardware.

Entre algunas de las diferencias de estos lenguajes es la seguridad. Un ejemplo de esto es al momento de declarar un "if", ya que la sintaxis obliga al desarrollador a poner una declaración por default y poner corchetes al principio y al final de una declaración de este tipo.

Otra gran diferencia que se puede observar en la Figura 4, es la cantidad de código y el número de archivos, ya que Apple decidió simplificar el lenguaje simplificando la sintaxis del código para que sea más fácil a la hora de escribir, más entendible y compilar con menos probabilidad de errores un proyecto. Por último, Swift incluye inferencia de datos, de esta forma el desarrollador no tiene que preocuparse por definir qué tipo de dato es una variable (Apple, 2014).

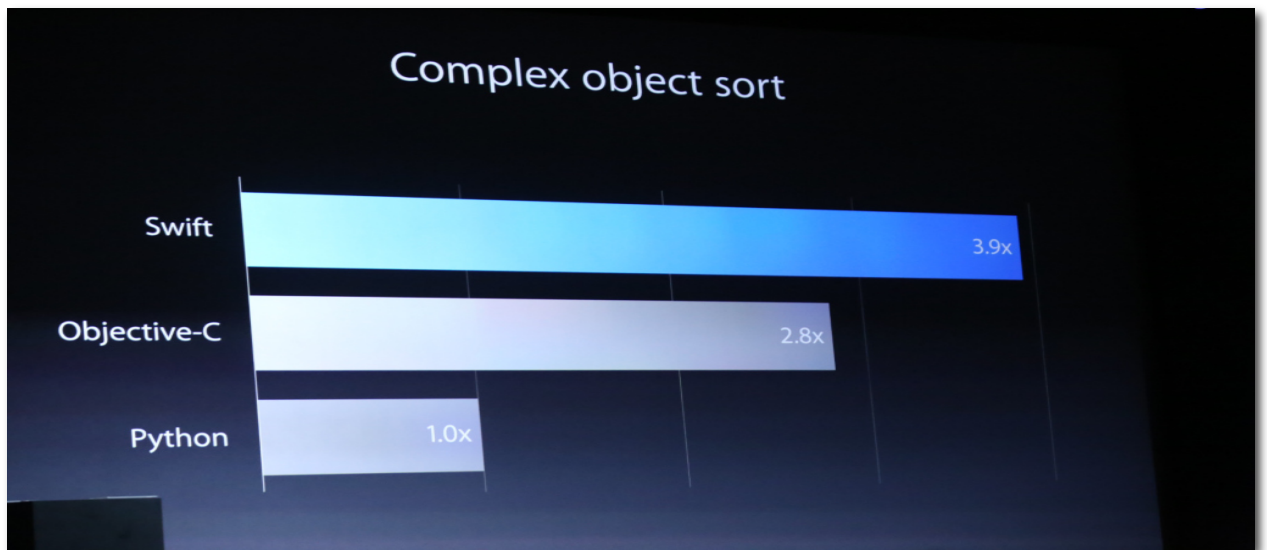


Figura 3. Rendimiento de Swift ante otros lenguajes de programación.

Objective-C vs Swift

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
    return 0;
}

println("hello, world")
```

Figura 4. Diferencia de sintaxis entre Objective-C y Swift.

PHP

Es un poderoso y sencillo lenguaje de uso gratuito y de código abierto que se ejecuta en un servidor web. Al ser un lenguaje de script no hay necesidad de compilar el código antes de usarlo. Fue diseñado principalmente para la creación de contenido web dinámico. Las principales ventajas que tiene al ejecutarse en un servidor web es que el resultado se incrusta en una página HTML, que es enviada al navegador. Este último no tiene conocimiento de esta operación realizada en el servidor, por lo que el código PHP está seguro. Esta operación permite realizar páginas con contenido dinámico al momento de realizar la petición de la página. Como se puede observar en la Figura 5, cuando se solicita un archivo PHP en el servidor web, el código PHP incluido con el documento HTML primero se ejecuta en el servidor. El resultado de esto se inserta en la página en lugar del código PHP y la página se reenvía al navegador mostrando contenido dinámico (Holzner, 2008).

MySQL

Es un sistema de administración de código abierto para bases de datos relacionales (RDBMS por sus siglas en inglés). Una base de datos es sencillamente una colección de archivos generalmente relacionados entre sí tales como texto, números y archivos que son almacenados y mantenidos en orden dentro de este sistema en archivos llamados bases de datos. MySQL usa un lenguaje de programación de consulta estructurado (SQL por sus siglas en inglés) el cuál está diseñado para manipular, gestionar y recuperar datos almacenados en bases de datos. Este software consiste de diferentes piezas, incluyendo MySQL server que es el que administra y ejecuta las bases de datos.

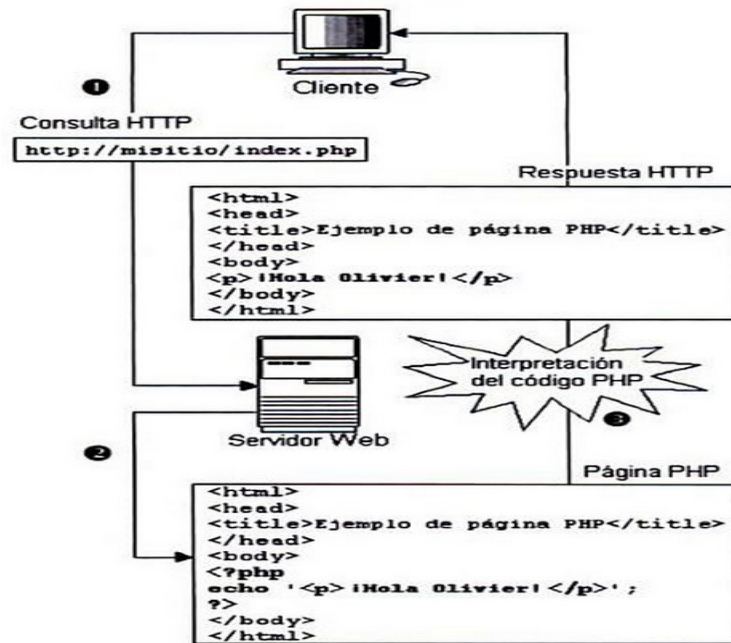


Figura 5. Solicitud de un cliente a un servidor web con PHP.

MySQL client que provee de una interface para el servidor y muchas otras utilidades que proveen mantenimiento, estabilidad y otras cosas. Una de las ventajas de MySQL es que puede ser usado con muchos lenguajes de programación como PHP. También, existen muchas aplicaciones que proveen de una interface más intuitiva a MySQL. Entre estas se encuentra PhpMyAdmin que se muestra en la Figura 6, el cual facilita su administración en servidores web al proveer una interface gráfica amigable (Ullman, 2006).

Server: localhost Database: base2 Table: persons

Structure Browse SQL Search Insert Export Operations Empty Drop

table personnes aaa; InnoDB free: 343040 kB; ('town_code') REFER 'base2:towns' ('town_code'); ('country_code') REFER 'base2:countries' ('country_code')

Field	Type	Attributes	Null Default	Extra	Action
<input type="checkbox"/> id	tinyint(4)		No	auto_increment	[edit] [delete] [add] [refresh] [undo] [redo]
<input type="checkbox"/> person_name	varchar(200)		No		[edit] [delete] [add] [refresh] [undo] [redo]
<input type="checkbox"/> town_code	varchar(5)		Yes 0		[edit] [delete] [add] [refresh] [undo] [redo]
<input type="checkbox"/> country_code	char(1)		Yes NULL		[edit] [delete] [add] [refresh] [undo] [redo]
<input type="checkbox"/> car_code	char(3)		No		[edit] [delete] [add] [refresh] [undo] [redo]

Check All / Uncheck All With selected: [edit] [delete] [add] [refresh] [undo] [redo]

Print view Relation view Propose table structure

Add field(s) At End of Table At Beginning of Table After id Go

Indexes:					Space usage:		Row Statistics:	
Keyname	Type	Cardinality	Action	Field	Type	Usage	Statements	Value
PRIMARY	PRIMARY	2	[edit] [delete]	id	Data	16,384 Bytes	Format	dynamic
town_code	INDEX	2	[edit] [delete]	town_code	Index	49,152 Bytes	Next Autoindex	5
country_code	INDEX	2	[edit] [delete]	country_code	Total	65,536 Bytes		
pays-ville	INDEX	2	[edit] [delete]	country_code town_code				

More than one INDEX key was created for column 'country_code'

Create an index on 1 columns Go

Figura 6. Herramienta Phpmysql en conjunto con MySQL para gestionar bases de datos de manera gráfica.

CAPÍTULO III

METODOLOGÍA

Características del proyecto

En este capítulo se describe los procesos y la metodología usados al momento de la planeación, diseño y desarrollo una aplicación nativa para iOS. Se describen las diferentes etapas generales del proyecto, las limitantes, los obstáculos y la forma en la cual estos fueron solucionados.

Cabe destacar que iOS es descrita como una de las plataformas más robustas, seguras y hasta cierto punto fáciles de programar, ya que Apple provee de mucha documentación de ayuda, además de herramientas para poder programar, ejecutar y probar el código sin la necesidad de tener un dispositivo iOS cerca. Muchos son los desarrolladores que, ya sea solo por curiosidad o porque encuentran interesantes y realmente útiles las herramientas que Apple provee y sus dispositivos, desarrollan aplicaciones para recibir algún ingreso monetario, asegurando el buen funcionamiento y distribución de sus aplicaciones.

El objetivo principal en el cual está enfocado este proyecto es en la creación de una aplicación nativa para iOS de enseñanza y que el material que este incluya, sea administrado de manera sencilla. Al momento de iniciar este proyecto se pensaba usar el lenguaje de programación Objective-C, debido a que Swift acababa de ser lanzado y aunque Apple también había lanzado junto con él documentación oficial de

ayuda, muchas de las cosas que se planeaban hacer, no se podían encontrar en tutoriales o foros de ayuda. Después de interactuar, manipular y ver la sintaxis reducida e intuitiva que tiene el código en Swift, se optó por escoger este lenguaje para el desarrollo de la aplicación nativa.

Para el desarrollo de este proyecto se utilizaron las siguientes tecnologías: (a) Mac OS X Yosemite 10.10.2, (b) Xcode en su versión 6.1, (c) simulador de iOS en su versión 8.2 (d) PHP para la realización de un portal web y (e) MySQL, gestor de base de datos que funciona en conjunto con PHP y HTML para mostrar contenido dinámico y guardar información.

Planificación

La presente investigación tiene como objetivo principal el aprendizaje sobre el proceso de desarrollo de una aplicación nativa para el entorno iOS con el lenguaje de programación Swift. Mientras se experimenta con estas tecnologías el objetivo principal es poder desarrollar una aplicación de enseñanza, en donde se puedan ver clases en video, tener apuntes y tareas asignadas en cada clase. Además de que el contenido pueda estar en un servidor web en donde pueda ser administrador por el instructor de la clase de manera sencilla.

Etapas del proyecto

La planificación de este proyecto se desarrolló en seis etapas principales: (a) selección de la plataforma móvil, (b) arquitectura de la aplicación, (c) gestión de contenido, (d) presentación de la información, (e) interacción alumno–instructor y, por ultimo, (f) implementación de tecnologías.

Selección de la plataforma móvil

Una plataforma móvil o sistema operativo, es la plataforma en donde hardware y el software de un dispositivo móvil trabajan y se comunican entre sí. Dependiendo del tipo de sistema con el que el dispositivo cuente, se consideran las herramientas y tecnologías a usar al momento de programar aplicaciones para estos.

Al momento de seleccionar la plataforma con la que se iba a trabajar, se consideraron tres factores principales:

1. Retorno de inversión: En otras palabras, lo que la plataforma ofrece a cambio del tiempo y recursos invertidos.

2. Herramientas disponibles: Entornos de programación, soporte, documentación y calidad de contenido.

3. Experiencia de aprendizaje: Lo que la plataforma ofrece en forma de conocimiento, experiencia y soluciones a situaciones del mundo laboral.

Al observar la Figura 7, se puede deducir fácilmente que las dos plataformas que actualmente dominan el mercado son el sistema desarrollado por Google, Android y el sistema operativo iOS desarrollado por Apple. Siendo Android el sistema que se encuentra instalado en la gran mayoría de los dispositivos.

Tomando al primer factor como referencia, el retorno de inversión, se puede observar en la Figura 8 que, aunque Android es el sistema que se encuentra instalado en una mayor cantidad de dispositivos, iOS domina la mayoría de los ingresos de las transacciones realizadas por medio de aplicaciones móviles, estas incluyen la compra de aplicaciones o del contenido extra dentro de estas.

Después de haber definido las dos plataformas móviles más relevantes, se

TOP U.S. SMARTPHONE OPERATING SYSTEMS BY MARKET SHARE

DURING Q1 2014

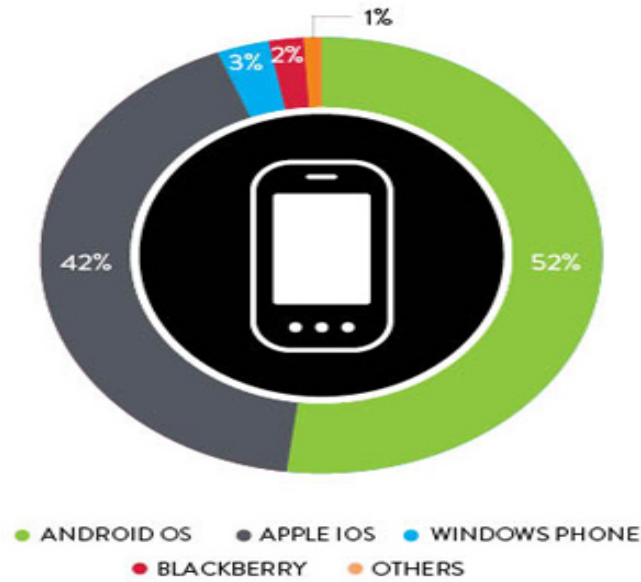


Figura 7. Sistemas operativos móviles más usados en 2014.

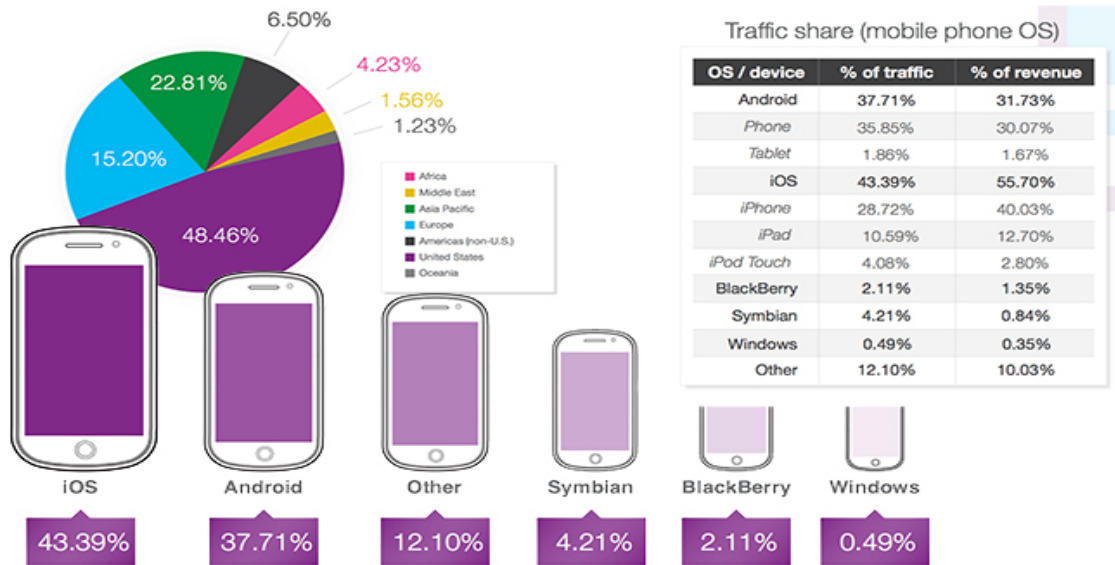


Figura 8. Retorno de inversión en cada plataforma móvil.

analizaron las herramientas de desarrollo de aplicaciones que cada una de estas plataformas tiene disponibles. Esto se hizo tomando como referencia el factor de herramientas disponibles e indagando más a detalle las características de cada uno de estos sistemas operativos.

Android es el nombre del sistema operativo desarrollado por Google, teniendo como una de sus características principales que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. Además de tener una muy buena integración con los servicios más populares de Google como el buscador, YouTube, mapas y Gmail, el desarrollo de aplicaciones para este sistema se realiza con un lenguaje de programación derivado de Java, el cual lleva por nombre Dalvik.

Además de que el sistema operativo provee todas las interfaces necesarias para el desarrollo de éstas aplicaciones como la cámara, el gps y otras funciones del dispositivo, existe una gran variedad de herramientas y ambientes de programación gratuitos. Android es un sistema de código abierto, eso quiere decir que los programadores pueden observar y modificar las aplicaciones para mejorarlas. Algo que destaca del desarrollo para Android, es que al ser libre, no es necesario pagar nada al momento de programar o probar la aplicación en un dispositivo con Android.

iOS es un sistema operativo desarrollado por Apple derivado de su sistema para equipos de escritorio Mac OS, para todos sus dispositivos móviles como el iPad, el iPhone, el iPod touch y el Apple TV. Una de las principales características es que es un sistema de arquitectura cerrada, por lo que solamente puede ser instalado y ejecutado en hardware de Apple. Desde un principio existen requisitos muy específicos para poder empezar a desarrollar aplicaciones para iOS y Mac OS. Se necesita

una computadora Macbook con un procesador Intel que pueda ejecutar por lo menos la versión de Mac OS X 10.9 Mavericks o la versión más reciente, Yosemite 10.10. En dado caso de no contar con hardware de Apple, existen servicios en línea que permiten usar máquinas virtuales con este sistema.

Apple es un poco ambigua al momento de proveer información para convertirse en un desarrollador de aplicaciones para iOS. Existe una gran cantidad de cosas que se pueden realizar de manera completamente gratuita. Entre ellas se encuentran el poder descargar el iOS SDK y ejecutar código en un simulador que viene incluido con este. Apple tiene una herramienta exclusiva llamada Xcode, que es la herramienta que contiene todo lo necesario para programar aplicaciones para su plataforma.

Apple ha hecho un excelente trabajo al incluir una gran cantidad de documentación de ayuda, recursos visuales y herramientas dentro de este entorno de desarrollo, por lo que aunque desarrollar aplicaciones para iOS no es del todo fácil, Xcode hace un excelente trabajo al momento del desarrollo. Existen características que no pueden ser ejecutadas en un simulador por las limitantes de hardware (como el acelerómetro y el gps), son funciones que solamente pueden ser visualizadas en un dispositivo de Apple que esté previamente dado de alta usando una licencia de desarrollador. Los desarrolladores inscritos al programa, además de poder vender sus aplicaciones en la tienda oficial y poder probar las aplicaciones en sus dispositivos, suelen recibir beneficios extras como tutoriales en video o recursos exclusivos. También se les da el privilegio de probar sus aplicaciones en las versiones beta de la extensa variedad de software que Apple posee.

Después de revisar la información y obtener los requisitos de cada sistema,

tomando como referencia el factor aprendizaje, se llegó a la conclusión de que android, es un sistema que destaca por todas las facilidades que ofrece para poder desarrollar dentro de esta plataforma, por lo que suele ser la más adoptada por los desarrolladores. iOS, aunque tiene muchos requisitos, también ofrece una gran cantidad de herramientas que facilita el aprendizaje y desarrollo de aplicaciones para su plataforma, además de tener la oportunidad de aprender sobre las tecnologías que son utilizadas en el proceso.

Después de indagar, valorar ambas plataformas y revisar los requisitos de cada una y las herramientas que ofrecen, se decidió desarrollar en iOS, dado que se cuenta con lo necesario, agregado al interés que existe en conocer más a fondo sus herramientas de desarrollo.

Arquitectura de la aplicación

Con la creciente demanda y producción de dispositivos móviles con distintas funciones, tamaños y sistemas operativos, el éxito de una aplicación depende en gran manera de poder llevar al usuario un producto no solamente funcional, sino que sea lo suficientemente flexible para poder ejecutarse en una gran cantidad de dispositivos con diferentes características y tamaños. Con la rápida proliferación de dispositivos móviles con características más avanzadas, también se han realizado avances en las tecnologías usadas para el desarrollo de aplicaciones. Actualmente y gracias al constante avance de estas tecnologías, hoy en día existen tres arquitecturas principales: (a) web apps, (b) aplicaciones híbridas y (c) aplicaciones nativas.

Las aplicaciones nativas, son aquellas que “viven” y funcionan dentro del ecosistema del dispositivo. Estas aplicaciones generalmente son instaladas por medio

de las tiendas de aplicaciones oficiales del sistema operativo. Están desarrolladas específicamente para una plataforma y tienen la ventaja de disponer de todos los recursos de hardware del dispositivo donde son instaladas. Estas aplicaciones a menudo integran gestos o funciones que adoptan del sistema operativo. La gran mayoría se pueden utilizar sin conexión a internet y pueden hacer uso de las notificaciones integradas en el dispositivo.

Las web apps o aplicaciones en web, en realidad no son del todo aplicaciones, pero están diseñadas para ofrecer la funcionalidad e interface de una aplicación nativa. Estas aplicaciones, fueron desarrolladas debido al alto tráfico generado por usuarios desde una gran variedad de dispositivos, por lo que una aplicación que no dependiera del hardware del dispositivo, fue la solución más rápida para poder llevar una misma aplicación a una extensa variedad de dispositivos. Estas aplicaciones dependen del entorno del equipo. Pueden ser ejecutadas solamente desde el navegador del dispositivo al conectarse a internet, aunque una gran cantidad cuentan con tecnologías que les permiten guardar temporalmente datos para poder ser usados sin conexión a internet.

Las aplicaciones híbridas como su nombre lo dice, son aplicaciones nativas que incorporan parte de las aplicaciones en web. Estas aplicaciones al igual que las nativas, se pueden descargar de la tienda oficial del sistema operativo del dispositivo correspondiente y pueden hacer uso de las características físicas del mismo. Así mismo, como las web apps, se basan en contenido web renderizado en un navegador, solo que en este caso, el navegador está envuelto dentro de la aplicación nativa.

En la Figura 9 podemos observar la comparación entre las diferentes tecnolo-

gías y sus características.

La idea principal del proyecto, es realizar una aplicación que pueda ser descargada de la tienda de aplicaciones, además de poder utilizar los recursos del equipo. Por lo que se llegó a la conclusión de que lo más ideal fuera diseñar una aplicación nativa.

Dentro de la arquitectura de una aplicación móvil, se encuentra el lenguaje de programación que se va a usar para diseñarla. Apple cuenta con dos lenguajes de programación para la realización de aplicaciones para iOS y Mac OS.

El primero se le conoce como Objective-C, el cuál es un lenguaje orientado a objetos basado en C y es el lenguaje más usado a la hora de programar aplicaciones de iOS y Mac OS.



Figura 9. Características de las diferentes arquitecturas de aplicaciones móviles.

Swift por su parte, fue recientemente lanzado como una versión más moderna de Objective-C. Aunque Objective-C es un poderoso lenguaje de programación, es un poco lento y complicado para los que no tienen mucha experiencia programando. Swift tiene entre sus ventajas, poder ser usado a la par con Objective-C, tener una sintaxis reducida, tener un desempeño más óptimo e incluir características de lenguajes de programación más actuales.

Tomando como referencia las diferencias entre ambos lenguajes de programación mencionados en el capítulo II. Se optó por utilizar Swift como lenguaje de programación principal para realizar el proyecto.

Gestión de contenido

En una aplicación móvil, algo muy importante después de un buen diseño es el contenido. El contenido enriquece en gran manera la experiencia que un usuario recibe al usar una aplicación. En un principio, cuando las computadoras y los dispositivos móviles no eran tan modernos, la memoria de almacenamiento de los equipos solía ser muy reducida, por lo que no había lugar a contenido de gran tamaño. Al avanzar la tecnología, la capacidad de almacenamiento de los dispositivos aumentó drásticamente. De igual forma, el contenido diseñado para este tipo de dispositivos tales como los sistemas operativos, los programas ejecutables, los videos y las imágenes fueron aumentando sus capacidad y por lo tanto, su tamaño. La mayoría de los dispositivos móviles hoy en día, aunque suelen tener un nivel de almacenamiento ideal para el usuario promedio, es común que con la descarga de aplicaciones móviles de gran tamaño como lo son los juegos o la captura de fotografías y video, lleguen a quedarse sin mucho espacio para almacenar información, por lo que es muy

frecuente que los dispositivos suelen alcanzar su límite máximo de almacenamiento, obligando al usuario a eliminar información valiosa.

Para dar solución a este problema, existen las aplicaciones de streaming, las cuales solamente tienen como requisito descargar una aplicación que no ocupa mucho espacio de almacenamiento, puesto que el contenido se encuentra en la nube todo el tiempo. El streaming que realizan estas aplicaciones es un proceso que se divide en cuatro partes: (a) solicitud de un archivo con un servidor remoto, (b) almacenamiento del archivo solicitado en un buffer o almacén temporal, (c) inicio de la reproducción, y (d) reproducción del buffer a pesar de no existir una conexión.

Existen muchos ejemplos de aplicaciones que utilizan esta tecnología, estando entre las más populares la aplicación de YouTube, una aplicación con su propio cliente web con videos subidos por diferentes usuarios. Spotify, un servicio de streaming de música y Netflix, en donde se pueden visualizar películas y series. Además del servicio de streaming, Spotify, tomando en cuenta que existe la posibilidad de no contar con una conexión a internet todo el tiempo, provee una opción para escuchar música sin necesidad de conexión a internet. Lo interesante de esta aplicación es que los archivos de música que se descargan, solamente pueden ser reproducidos dentro de la aplicación, puesto que Spotify solamente permite al usuario escuchar la música, sin la opción de guardarla en su computadora ni en otra parte del dispositivo.

Después de analizar las ventajas que este tipo de aplicaciones proveen, se decidió usar esta tecnología de streaming en la aplicación tomando como ejemplo a la aplicación Spotify, para que de esta forma la aplicación sea ligera y no ocupe mucho espacio dentro de la memoria del dispositivo, al momento de descargar el conte-

nido, este solamente se pueda reproducir dentro de la aplicación.

Presentación de la información

Un elemento muy importante en la enseñanza, es la forma en la cual se imparte el conocimiento, por lo que para que una aplicación con contenido didáctico pueda ser efectiva, tiene que presentar de una manera precisa e interesante la información.

Para el diseño de la presentación del contenido, se analizó la forma en la que actualmente se basa el sistema de enseñanza presencial. Si se observa la Figura 10, este tipo de educación consiste de tres cosas básicas:

1. El profesor prepara e imparte el material docente para la clase.
2. Los alumnos escuchan al maestro en clase y toman apuntes.
3. El profesor les asigna una tarea para que el estudiante pueda evaluar el conocimiento adquirido.

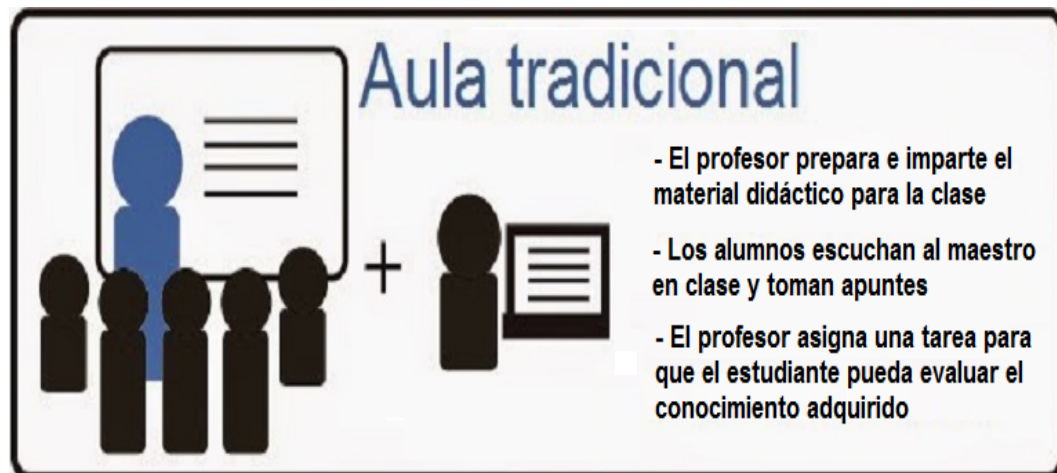


Figura 10. Características del sistema de enseñanza presencial.

Tomando en cuenta lo anterior, se planteó que la estructura de la aplicación, intentara imitar ese sistema para que el alumno se sienta más familiarizado con la aplicación al usar este mismo sistema de aprendizaje. Se plantearon tres secciones y quedarían de la siguiente manera:

1. El rol del profesor sería representado en forma de contenido audiovisual, por lo que en este caso se decidió que lo más adecuado sería impartir las clases por medio de videos, en donde el profesor se muestre explicando diferentes tópicos referentes a su material didáctico, puesto que en un modelo de enseñanza real, el maestro explica los temas de forma general con los alumnos por medio de palabras, texto e imágenes.

2. Siguiendo con este sistema de enseñanza, se decidió incluir una sección en donde el profesor pueda agregar apuntes extra a la clase por medio de un documento escrito, que sirva como apuntes y también como un resumen del contenido más importante. Para finalizar, se colocaría una sección en donde el profesor incluyera una actividad o tarea para reforzar lo aprendido en la clase.

Interacción alumno-instructor

Una de las cosas más importantes al momento de la enseñanza, es la interacción que existe entre el alumno y el instructor. El alumno tiene la tarea de escuchar la clase que el maestro está impartiendo. Anotar todo el contenido de la clase a manera de apuntes para resumir lo más importante y al final, demostrar lo aprendido por medio de alguna actividad que el maestro decida aplicar. Por su parte, el maestro tiene la responsabilidad de preparar no solo las clases y el material que va a impartir, sino también el tiempo de cada clase y las fechas en las que estas van a ser dadas.

Además de incluir material de ayuda por si el alumno tiene dudas y evaluar por medio de tareas el conocimiento impartido al alumno.

Para que la aplicación no rompa con este sistema, como se puede ver en la Figura 11 se llegó a la conclusión de implementar dos interfaces:

1. La primera interface de interacción estaría representada por la aplicación móvil, en donde el alumno podrá mirar las clases, repasar los apuntes y realizar las actividades de evaluación.

2. La segunda interface estaría representada por un portal web, en el cuál el maestro puede manipular de la forma que sea necesario las clases. El maestro podría preparar sus clases, hacer un documento para cada clase en donde se encuentren resaltados los puntos más importantes y un documento con la actividad de evaluación para el alumno. El maestro podrá subir y borrar las clases cuando crea que es necesario.

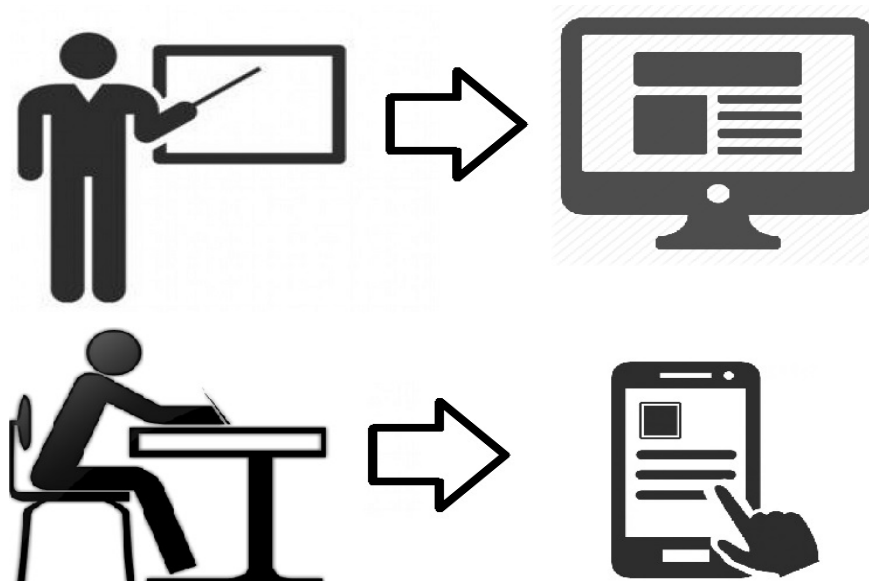


Figura 11. Interfaces definidas para la interacción de alumno y maestro.

La idea al momento de implementar estas dos interfaces, es que ambas partes puedan interactuar sin ninguna dificultad y de manera intuitiva. Estas interfaces tienen la intención de proveer de un ambiente familiar de aprendizaje para que tanto el maestro como el alumno, puedan realizar sus actividades independientemente uno del otro.

Implementación de las tecnologías

Después de haber pasado por todas las fases de planeación de este proyecto, se estableció de manera concreta el tipo de tecnologías e interfaces a implementar. Se utilizó la herramienta de desarrollo Xcode para hacer la parte visual y lógica de la aplicación, creando un proyecto con los temas base que Xcode provee.

Durante el inicio de la fase de desarrollo de la aplicación, se estableció que era necesario darle el control al instructor en la forma que imparte sus clases. Se tomó la decisión de implementar un servidor web con una interface en Bootstrap 3 para el aspecto visual del portal. PHP y MySQL para los procesos, la gestión del contenido didáctico y la base de datos en donde estarán almacenados los archivos. Estas tecnologías serían usadas para crear un pequeño sistema con una página de inicio, requiriendo usuario y una contraseña previamente establecido. Después de que el maestro ingresara su usuario y contraseña, pasaría a almacenar dentro de una base de datos en el servidor web el archivo de la clase en video con un formato mp4, los manuales y las tareas con formato PDF en sus correspondientes secciones: (a) clase, (b) manual y (c) tarea.

Esto permitiría hacer que la aplicación ya terminada fuera más ligera y tomara

menos tiempo al usuario descargarla, ya que todo el contenido del responsable de impartir el curso se encontraría en la nube. Tomando en cuenta la posibilidad de que el usuario no contara con una conexión a internet óptima o disponible la mayor parte del tiempo, se tomó la decisión de incluir dentro de la sección de cada clase, una opción que permitiera al usuario descargar los archivos de la clase sin internet, dentro de la aplicación del dispositivo. Los archivos descargados estarían almacenados dentro de la aplicación, evitando que puedan ser guardados dentro del dispositivo.

Aunque darnos de alta como desarrolladores en la página de Apple no es del todo necesario. bajar e instalar Xcode si lo es, ya que aquí es en donde se desarrollan las aplicaciones para iOS y Mac OS. Para poder descargar Xcode, como se muestra en la Figura 12, es necesario ir a la Mac App Store. Dentro de la tienda de aplicaciones en la sección de buscar, solamente se tiene que colocar el nombre “Xcode” y listo. Después de que la aplicación sea descargada, puede ser ejecutada, después de configurar lo básico, ya se puede empezar a programar.

Como se puede observar en la Figura 13, para la realización de este proyecto, se tomó como base el tema “Single View Application” disponible en Xcode y la opción de proyecto en lenguaje Swift. Después de haber seleccionado estas opciones, Xcode nos provee rápidamente de una interfaz visual que puede ser editada fácilmente.

Dentro del proyecto de la aplicación, se utilizó el archivo de storyboard que Xcode generó del proyecto para conectar las diferentes ventanas y de esta forma definir la forma de interacción entre cada vista. Primero se agregó una pantalla principal en donde las clases serían desplegadas en orden. Después de la vista principal, al

momento de seleccionar la clase, se llevaría a otra ventana en donde se desplegarían tres opciones: (a) clase, (b) manual y (c) tarea. Además de estas opciones, se pueden observar dos opciones extras en la cabecera de la aplicación. En la esquina superior izquierda, al momento de ejecutarse la aplicación aunque no se observa en la imagen, al ejecutar la aplicación se muestra un botón para regresar al menú principal donde se encuentran las clases. En la esquina superior derecha de la vista detallada de la clase, se puede observar un ícono de que indica que la clase puede ser descargada para verse sin conexión a internet (ver Figura 14).



Figura 12. Pasos para descargar Xcode en Mac OS.

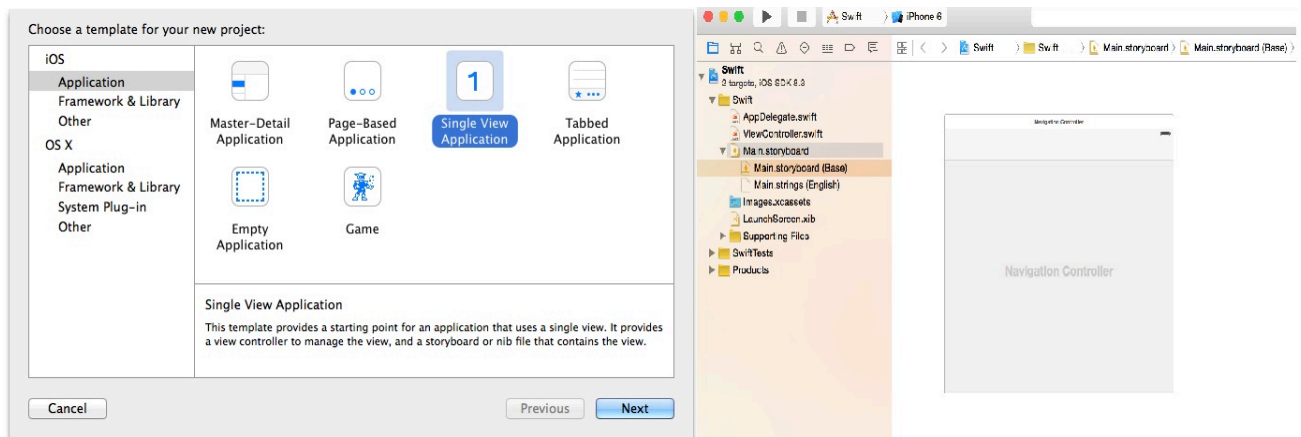


Figura 13. Creación de la aplicación, archivos base y storyboard en Xcode.

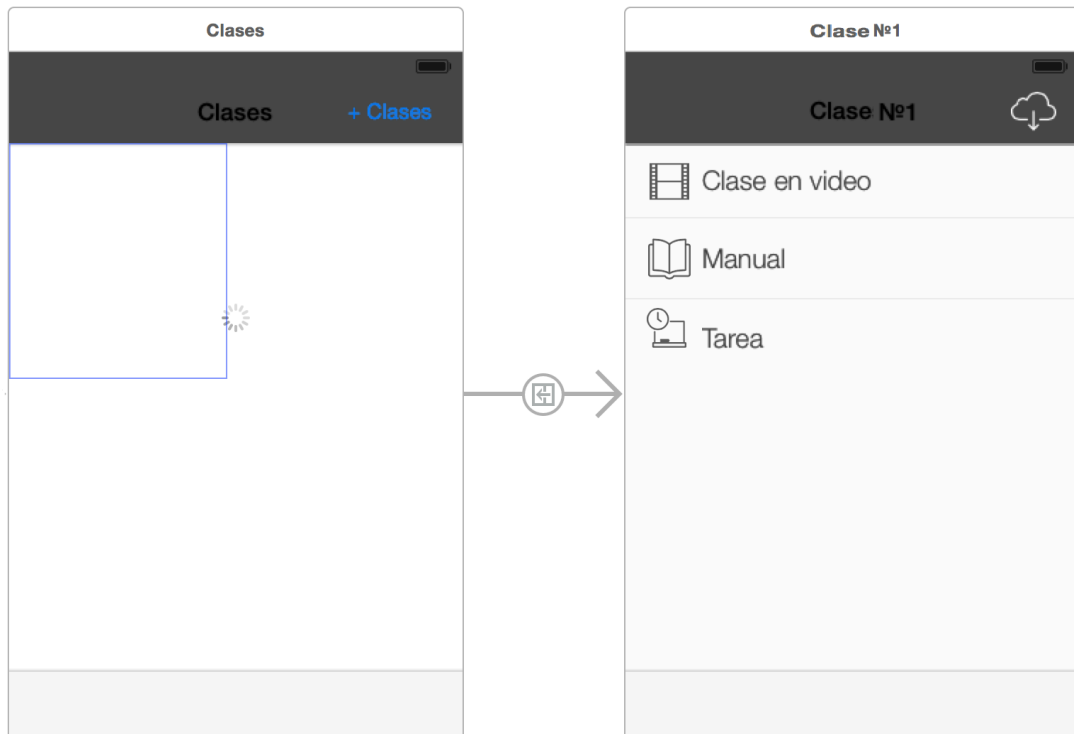


Figura 14. Pantalla principal y menú al seleccionar una clase.

Al momento de seleccionar la clase en video como se observa en la Figura 15, aparece una ventana que le hace saber al usuario que el video está en proceso de reproducirse y la opción de cancelar la reproducción regresando al menú anterior.

Al seleccionar la opción “Manual”, se despliega una ventana en donde el documento PDF se despliega. Se puede observar que en la esquina superior izquierda se incluyó una opción en donde se puede ver el video de nuevo sin tener que regresar a la vista detallada de la clase dentro de la aplicación (ver Figura 16).

En forma muy similar, en la Figura 17 se muestra que al seleccionar la opción de “tarea”, se despliega el documento correspondiente, pero sin la opción de mirar la clase de nuevo.

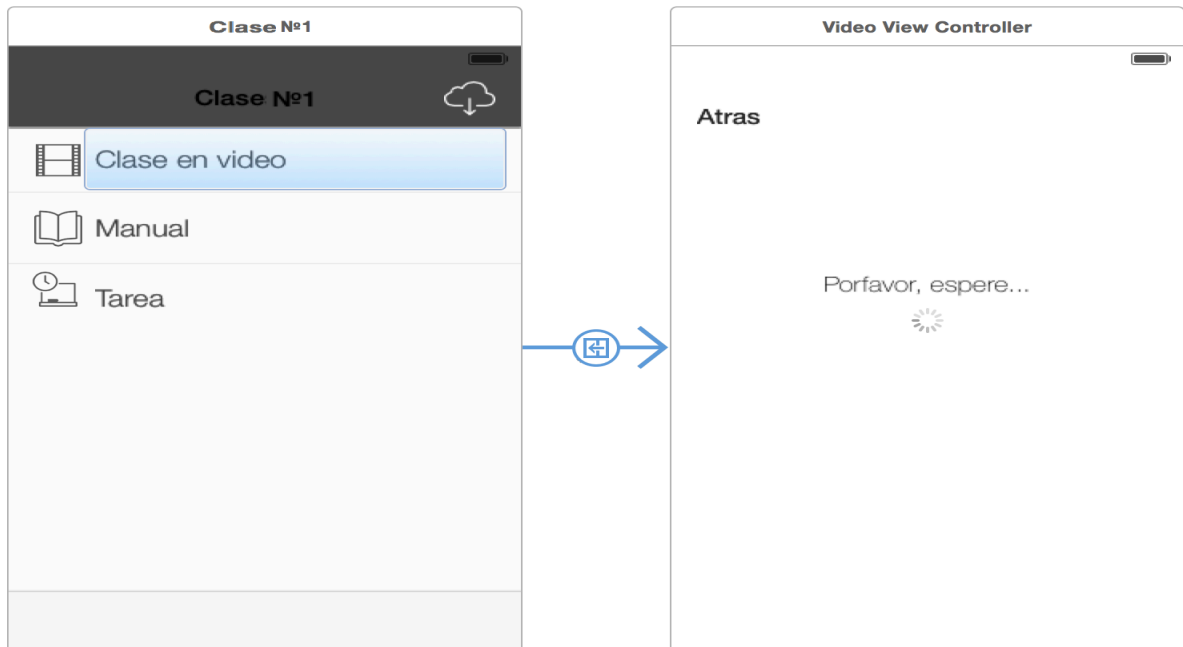


Figura 15. Vista principal al seleccionar la opción de video de una clase.

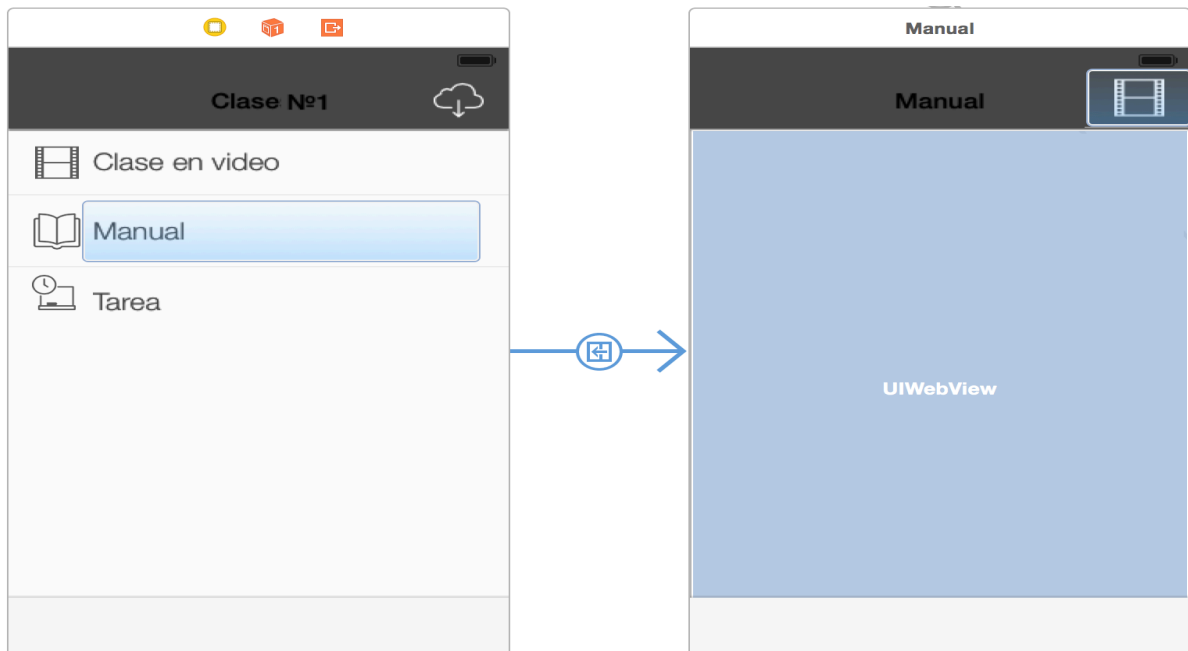


Figura 16. Pantalla principal al seleccionar la opción "manual" dentro de la clase.

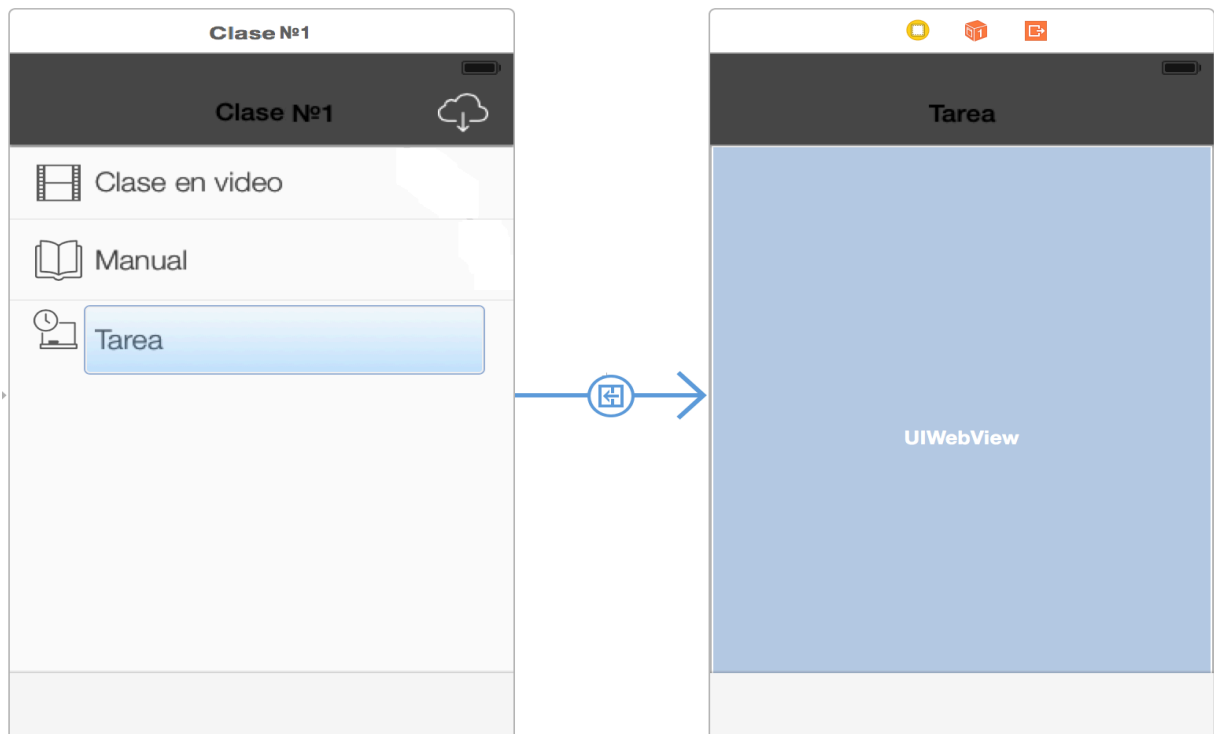


Figura 17. Pantalla principal al elegir la sección de tarea en una clase.

Después de tener las vistas y la forma en la que la aplicación iba a mostrar la información, el paso siguiente fue el diseño del portal web. Se implementaron 3 vistas principales para el portal web. En la Figura 18 se muestra la primera vista y consiste en una página de inicio en donde se le pide proporcionar un usuario y una contraseña previamente definidos en el momento de la programación. Después de autenticarse, en la segunda vista se muestra una pantalla con las clases que actualmente se encuentran en la base datos con la opción de poder mirar tanto el archivo de video de la clase, como los documentos incluidos en las secciones de manual y tarea. Por último, en la tercera vista del portal web, se incluye una sección en donde el maestro puede subir el contenido de su clase en cada lugar correspondiente.

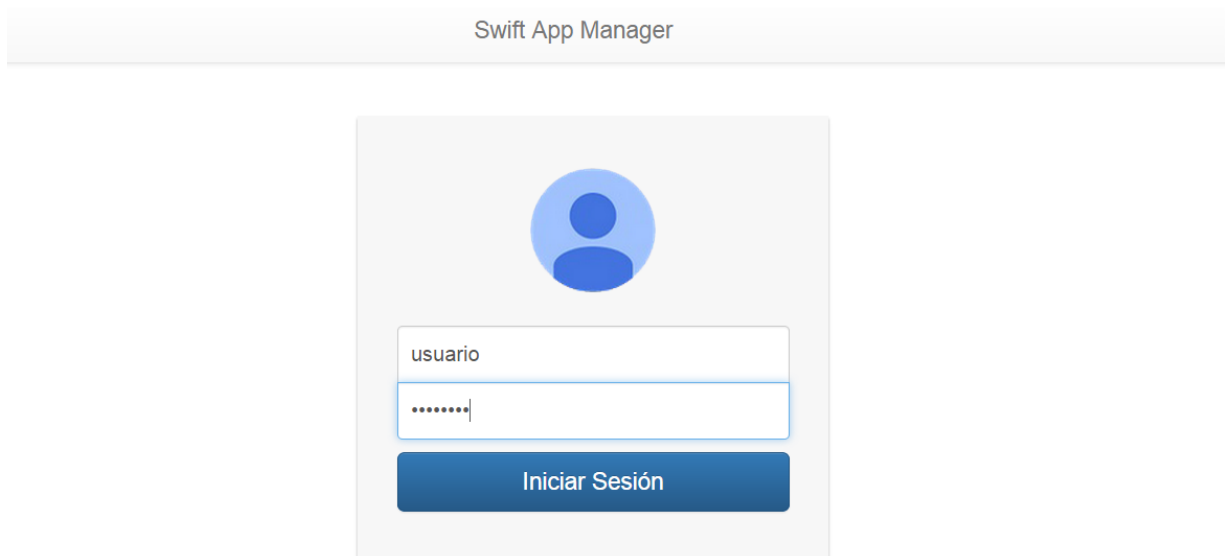


Figura 18. Vista de inicio de sesión del portal web del maestro.

Para realizar la conexión entre la aplicación y los archivos del servidor, se utilizó una tecnología llamada JSON. La tecnología de Notación de objetos de JavaScript (JSON, por su abreviatura en inglés) es a grandes rasgos, un formato para el intercambio de datos. Una de las ventajas de JSON es que puede ser interpretado por casi cualquier lenguaje de programación, siendo esta la opción más adecuada para este proyecto. Lo que decidió fue usar JSON en los archivos PHP del servidor web para poder transmitir la información de los archivos almacenados en la base de datos. En la Figura 19 se describe gráficamente el proceso de comunicación entre la aplicación y el portal web usando JSON.



Figura 19. Interacción entre la aplicación y el servidor web al momento de intercambiar datos.

La Figura 20 incluye un fragmento de la función en el archivo clases.php que obtiene un arreglo con la información de los archivos de la base de datos en un arreglo para darle formato con JSON.

Después de finalizar el portal web y pasar la información de los archivos en la base de datos a JSON, se estableció la conexión para ser obtenida por la aplicación. Para usar la información en JSON se utilizó un framework llamado Alamofire, el cual permite hacer peticiones JSON a servidores web. Para poder visualizar el contenido y agregar la opción para descargarse en dado caso de que se quiera acceder al material sin conexión a internet, se utilizó un framework llamado SQLite, el cual crea una base de datos temporal en la aplicación.

Como se puede observar en la Figura 21, se incluye un fragmento del código en donde se le dice a la aplicación el servidor donde se encuentran los archivos y el nombre de la base de datos. También se puede observar un fragmento del archivo en donde se importan los frameworks Alamofire y SQLite al proyecto.

```
header("Content-Type: application/json");

if (isset($_POST['from']) && !empty($_POST['from']))
{
    $from = clear($_POST['from']);
    if ($from == "app"){
        echo json_encode(array('result' => getClases()));
    }
    else
    {
        header("Location:../");
    }
}
else
{
    echo json_encode(array('result' => "0"));
}
```

Figura 20. Función PHP que traduce la información de la base de datos a JSON.

Alamofire se encarga de hacer la petición de datos JSON que fueron recibidos de la aplicación web para mostrarlos en la interfaz de la aplicación móvil. El framework SQLite almacena los archivos por el tiempo que dure la sesión del usuario o por tiempo indefinido, si el usuario utiliza la opción de descargar el contenido y tenerlo disponible sin necesidad de conexión a internet. Al final de la implementación, la aplicación en Xcode fue compilada y ejecutada en el simulador.

```

import UIKit

class Const: NSObject {

    class func getPath() -> String {
        return "http://www.appswift.dreamhosters.com/"
    }

    class func getNameDB() -> String {
        return "swift"
    }

    class func getTblNameOfMain() -> String {
        return "Clases"
    }

    class func successDownloadString() -> String {
        return "Ok"
    }

    class func isDownloadString() -> String {
        return "Descargado"
    }

    class func errorWithFirstStartApp() -> String {
        return "Se necesita internet para iniciar la app..."
    }
}

import UIKit
import Alamofire
import SQLite

let clases_tbl = localDB[Con.getTblNmOfMain()]

let id = Expression<String>("id")
let num = Expression<String>("num")
let nombre = Expression<String>("nombre")
let srcClase = Expression<String>("srcClase")
let srcLibro = Expression<String>("srcManual")
let srcTarea = Expression<String>("srcTarea")

if Con.isConnectedToNetwork() == true {

    Alamofire.request(.POST, Con.getPath()+"/Assets/Clases.php", parameters: ["from":
'app'])
        .responseJSON { (_, __, JSON, _) in
            var d = JSON as! NSDictionary
            var les = d["result"] as! NSArray

            var replace:Bool = false
            if les.count != lectures_tbl.count {
                lectures_tbl.delete()!
                replace = true
            }
        }
}

```

Figura 21. Archivos Swift que traducen la información de JSON para visualizarla en la aplicación y almacenarla temporalmente.

Al momento de ejecutar la aplicación, las mismas clases que se encuentran dentro de la plataforma web se visualizan con su información correspondiente dentro de la aplicación. Todo esto se muestra de la forma deseada, dependiendo del dispositivo donde sea ejecutada. El portal web funciona de la misma manera, con la excepción de que no está optimizado para móviles, puesto que su uso se encuentra reservado para dispositivos de escritorio.

CAPÍTULO IV

CONCLUSIONES

Resumen

En el capítulo I se destaca la relevancia del rol que juegan en el mundo actual las tecnologías móviles y el cómo han cambiado la forma de comunicarse, consultar y compartir información. Se presentan objetivos: (a) explorar el desarrollo de aplicaciones móviles educativas, (b) aprender a programar para iOS usando el nuevo lenguaje Swift y (c) desarrollar una aplicación nativa y un portal web para administrar el contenido.

El capítulo II se enfoca de lleno en las características principales del m-learning y en sus fases esenciales que son: (a) el enfoque a dispositivos móviles, (b) el aprendizaje fuera del aula de clases y (c) la flexibilidad del contenido el tiempo para el estudiante. Se habla de las posibilidades que ofrecen las aplicaciones educativas para la enseñanza como foros de discusión, compartir logros, notificaciones locales o remotas. Se describen la necesidad de los sistemas operativos Mac OS, iOS a la hora de desarrollar aplicaciones. Se habla del entorno de desarrollo Xcode y del lenguaje de programación Swift. Al final se mencionan las principales características de PHP, MySQL y el uso de ambos en servidores web.

En el capítulo III se describe a detalle el proceso de planeación, diseño e implementación de los elementos de la aplicación nativa para iOS de enseñanza. Se mencionan las 6 etapas principales de la planeación, las cuales fueron: (a) selección

de la plataforma móvil, (b) arquitectura de la aplicación, (c) gestión de contenido, (d) presentación de la información, (e) interacción alumno–instructor y, por último, (f) implementación de tecnologías. Cada una de estas etapas contiene una explicación sobre las diferentes opciones que se consideraron y al final de cada una, la decisión tomada en base a la información analizada. Se habla del proceso de implementación de las tecnologías, tomando como punto de partida el diseño de la aplicación usando la herramienta Xcode, las diferentes herramientas utilizadas para la gestión del contenido y las tecnologías para desarrollar el portal web en donde el instructor es el encargado de gestionar el contenido de cada clase.

En el capítulo IV se mencionan recomendaciones al momento de la realización del proyecto, las conclusiones y los resultados obtenidos.

Conclusiones del proyecto

El documento presenta un proyecto de software enfocado a cumplir el objetivo de la enseñanza usando el m-learning y tecnologías para el desarrollo de aplicaciones con este fin. El resultado es una aplicación nativa de iOS que permite visualizar contenido educativo desarrollado por el responsable del curso o clase, además de que cada clase incluye tres secciones. La primera sección es la clase en video que el estudiante puede ver. La segunda y tercera sección consisten en documentos con formato PDF con apuntes, guías para entender mejor la clase y al final una tarea para reforzar lo visto en la clase. Las clases y el contenido pueden ser administrados enteramente por el educador por medio de un portal web que permite subir o eliminar las clases, manuales y tareas.

Una de las principales características del proyecto, es que todo el contenido

está almacenado en la nube, por lo que no ocupa espacio físico en el dispositivo. Si no se cuenta con conexión a internet, puede ser descargado dentro de la aplicación y no en el dispositivo. Esto permite en cierta forma, proteger el contenido en dado caso que el instructor no desee que este sea descargado o no se tenga permisos para distribuir libremente el contenido ya sea por parte del instructor o del alumno.

Swift es un lenguaje que facilita en gran manera el aprendizaje en la programación de aplicaciones para iOS. Por ser un lenguaje recientemente lanzado al público, al momento de buscar documentación sobre temas más avanzados, suelen aparecer artículos relacionados a celebridades o de un software financiero usado para la conexión entre bancos. Aunque es mucho más intuitivo que Objective-C, esto no significa necesariamente que se puede alcanzar un nivel de programación avanzado en poco tiempo.

Resultados

El resultado principal de este proyecto conlleva a una visión más amplia, al momento de desarrollar aplicaciones nativas con contenido educativo, en donde este mismo contenido puede ser administrado por uno o varios instructores por medio de un portal web. Este mismo tipo de aplicaciones pueden ser usadas no solamente con fines didácticos, sino también adaptarse a diferentes situaciones. Se logró obtener conocimiento puntual y funcional sobre el m-learning, tecnologías de desarrollo web, de aplicaciones móviles y la capacidad de mezclar estas tecnologías para funcionar en conjunto al momento de resolver un problema real.

CAPÍTULO V

TRABAJOS FUTUROS

A continuación se mencionan los planes para el desarrollo de trabajos futuros:

1. Incursionar más profundamente en temas avanzados de Swift para crear aplicaciones más robustas y con una mayor variedad funciones.


2. Adquirir una licencia de desarrollador de Apple para la correcta ejecución de las aplicaciones desarrolladas en un dispositivo iOS. Para agregar funciones extra como las notificaciones remotas, el acelerómetro, la cámara. Al final se espera colocar la aplicación para su descarga en la tienda oficial de aplicaciones de Apple.

3. Reestructurar el contenido dentro de la aplicación para mostrar las clases dentro de una sección de cursos. De esta forma será posible tener diferentes cursos simultáneamente dentro de la misma aplicación sin borrar clases anteriores.

4. Incluir en la aplicaciones funciones más complejas como preguntas o exámenes al final de cada clase o curso. Agregar foros discusión, tener un formulario de contacto para cualquier duda o pregunta, incluir un registro de usuarios y una lista de privilegios para llevar el control de cada estudiante y agregar transacciones o publicidad dentro de la aplicación.

APÉNDICE A

APLICACIÓN NATIVA IOS

Carrier 

3:25 PM



Game Center



Extras



Apple Watch



Swift Net



Safari

Carrier

4:48 AM



Done 0:07



-0:33





Clases

+ Clases



Clase Nº1

Introduccion y principios basicos de redes



Clase Nº2

TCP y UDP



C

Por el momento no tenemos cursos disponibles.



4

Es
Mo
p

s y
s de
ento

OK




Clase Nº5

Direcciones IP y mascararas de red

APÉNDICE B

PORTAL WEB

Swift App Manager



[Inicio](#) [Agregar clase](#) [Cerrar Sesión](#)

Swift App Manager

Clases actuales

ID	No.	Nombre	Video	Manual	Tarea	
1	Clase #1	Introduccion y principios basicos de redes	Ver video	Ver manual	Ver tarea	<input type="button" value="Borrar"/>
2	Clase #2	TCP y UDP	Ver video	Ver manual	Ver tarea	<input type="button" value="Borrar"/>
3	Clase #3	Estructura y Monitoreo de paquetes	Ver video	Ver manual	Ver tarea	<input type="button" value="Borrar"/>
4	Clase #4	Programas y herramientas de mantenimiento	Ver video	Ver manual	Ver tarea	<input type="button" value="Borrar"/>
5	Clase #5	Direcciones IP y mascaras de red	Ver video	Ver manual	Ver tarea	<input type="button" value="Borrar"/>

[Inicio](#) [Agregar clase](#) [Cerrar Sesión](#)

Swift App Manager

Agregar clase

Numero de clase

Nombre

Video (mov, mp4, m4v, 3gp)

 Ningún archivo seleccionado

Manual (pdf)

 Ningún archivo seleccionado

Tarea (pdf)

 Ningún archivo seleccionado

LISTA DE REFERENCIAS

- Ally, M. y Prieto, J. (2014). What is the future of mobile learning in education? *Revista de Universidad y Sociedad del Conocimiento*, 11(1), 142-151. doi:10.7238/rusc.v11i1.2033
- Apple. (2014). *iOS technology overview*. Recuperado de <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iOSTechOverview.pdf>
- Bakir, A. (2014). *Beginning iOS media app development*. Recuperado de <https://books.google.com.mx/books?id=7CqbBQAAQBAJ>
- Berge, Z. L. y Muilenburg, L. (2013). *Handbook of mobile learning*. Recuperado de <https://books.google.com.mx/books?id=OJrifm5uSGUC>
- Bidin, S. y Ziden, A. A. (2013). Adoption and application of mobile learning in the education industry. *Procedia - Social and Behavioral Sciences*, 90, 720-729. doi:10.1016/j.sbspro.2013.07.145
- Bucanek, J. (2013). *Learn IOS 7 app development*. Recuperado de https://books.google.es/books?id=W_JZAgAAQBAJ
- Clark, J. F. (2012). *History of mobile applications*. Recuperado de <http://www.uky.edu/~jclark/mas490apps/History%20of%20Mobile%20Apps.pdf>
- Conder, S. y Darcey, L. (2010). *Android wireless application development*. Recuperado de <http://books.google.com.mx/books?id=XqIMCZZjkjsC>
- Filgueira Gomis, J. M. (2014). *Mobile learning: Estrategias para el uso de aplicaciones, smartphones y tablets en educación*. Recuperado de <https://books.google.es/books?id=t9VsBAAAQBAJ>
- Hardy, J., Farzin, F. y Scanlon, M. (2013). *The science behind lumosity*. Recuperado de http://lumosblog.wpengine.netdna-cdn.com/wp-content/uploads/2013/11/The_Science_Behind_Lumosity_v2.pdf
- Keegan, D. (2005). *The incorporation of mobile learning*. Recuperado de <http://www.mlearn.org.za/CD/papers/keegan1.pdf>

- Kelly, M. y Nozzi, J. (2013). *Mastering Xcode: Develop and design*. Recuperado de <http://books.google.com.mx/books?id=tXwQAAAAQBAJ>
- Kim, B. (2013). The mobile shift. *Library Technology Reports*, 49(6), 5-8. doi:10.5860/ltr.49n6
- Landers, P. (2002). *From e-learning to m-learning*. Recuperado de http://learning.ericsson.net/mlearning2/project_one/leo.html
- Lang, P. (2009). *Edu: Technology and learning environments in higher education*. Recuperado de <https://books.google.com.mx/books?id=PCjLCKQBgkC>
- McQuiggan, S., McQuiggan, J., Sabourin, J. y Kosturko, L. (2015). *Mobile learning: A handbook for developers, educators, and learners*. Recuperado de <https://books.google.es/books?id=zpSDBgAAQBAJ>
- mobiForge. (2014). *Global mobile statistics*. Recuperado de <http://mobiforge.com/research-analysis/global-mobile-statistics-2014-part-a-mobile-subscribers-handset-market-share-mobile-operators#subscribers>
- Pachler, N., Bachmair, B. y Cook, J. (2009). *Mobile learning: Structures, agency, practices*. Recuperado de <https://books.google.com.mx/books?id=4wic3OISf0EC>
- Palfrey, J. y Gasser, U. (2013). *Born digital: Understanding the first generation of digital natives*. Recuperado de <http://books.google.com.mx/books?id=LIFNT1ER4scC>
- Pelletier, J. (2013). *Mobile app manual: The blueprint*. Recuperado de <https://books.google.es/books?id=kqXAUvj07TMC>
- Perry, D. (2003). *Handheld computers (PDAs) in schools*. Recuperado de [http://bee-it.co.uk/Guidance%20Docs/Becta%20Files/Publications/44.%20Research%20report%20Handheld%20computers%20\(PDAs\)%20in%20schools.pdf](http://bee-it.co.uk/Guidance%20Docs/Becta%20Files/Publications/44.%20Research%20report%20Handheld%20computers%20(PDAs)%20in%20schools.pdf)
- Phillips, M. (2011). *Connected learning*. Recuperado de <http://www.acu.edu/technology/mobilelearning/documents/research/phillips/phillips-final-report.pdf>
- Prensky, M. (2001). Digital natives, digital immigrants. *MCB University Press*, 9(5), 1-6.
- Ramos, A. I., Herrera, J. A. y Ramírez, M. S. (2010). Desarrollo de habilidades cognitivas con aprendizaje móvil: un estudio de casos. *Revista Científica de Educomunicación*, 17(34), 201-209. doi:10.3916/C34-2010-03-20

- Rekkedal, T., Dye, A., Fagerberg, T., Bredal, S., Midtsveen, B. y Russell, J. (2005). *Mobile learning: A practical guide*. Recuperado de http://www.ericsson.com/res/thecompany/docs/programs/incorporating_mobile_learning_into_mainstream_education/book.pdf
- Sadun, E. y Wardwell, R. (2014). *The core iOS developer's cookbook*. Recuperado de <https://books.google.com.mx/books?id=2sj6AgAAQBAJ>
- Salz, P. A. y Moranz, J. (2013). *The everything guide to mobile apps*. Recuperado de <https://books.google.es/books?id=ApWE0ruRV3gC>
- Sarwar, M. y Soomro, T. R. (2013). Impact of smartphone's on society. *European Journal of Scientific Research*, 98(2), 216-226.
- Statista. (2014). *Number of apps available in leading app stores as of July 2014*. Recuperado de <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>
- Statista. (2015). *Most popular Apple app store categories in March 2015, by share of available apps*. Recuperado de <http://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>
- Udell, C. (2014). *Mastering mobile learning*. Recuperado de <https://books.google.com.mx/books?id=kPlwBAAAQBAJ>
- Ullman, L. (2006). *MySQL, second edition: Visual quickstart guide*. Recuperado de <http://books.google.com.mx/books?id=WjIQh9bo7DoC>
- UNESCO. (2011). *UNESCO mobile week report*. Recuperado de <http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/ED/ICT/pdf/UNESCO%20MLW%20report%20final%2019jan.pdf>
- Varma, J. (2015). *Xcode 6 essentials*. Recuperado de <https://books.google.com.mx/books?id=SXNuBgAAQBAJ>
- Vesselinov, R., y Grego, J. (2012). *Duolingo effectiveness study*. Recuperado de http://static.duolingo.com/s3/DuolingoReport_Final.pdf
- Warner, R. y Privat, M. (2011). *Beginning OS X lion apps development*. Recuperado de <https://books.google.es/books?id=4phLLxdrRYQC>
- Zammetti, F. (2013). *Pro iOS and Android apps for business*. Recuperado de <https://books.google.es/books?id=IZbJAQAAQBAJ>