



**UNIVERSIDAD DEL ACONCAGUA**  
**Facultad de Ciencias Sociales y Administrativas**  
**Ingeniería en Software**

**Smartphones:  
Aplicación móvil académica de la  
Universidad del Aconcagua**

**Daniel Gómez Sevald**

**Lejajo: 17301**

**Año: 2013**

**Directora de Tesis: Prof. Rosana Giménez**

## **Calificación**

## **Resumen Técnico**

El rápido avance de la tecnología desde épocas no muy distantes, ha creado nuevas “necesidades” en la comunidad con respecto a los servicios de telecomunicaciones. La conexión permanente por telefonía celular ha llegado a ser un elemento tan indispensable para la sociedad que ha logrado desarrollar incluso una dependencia en el consumidor.

Este nuevo estilo de vida presiona a las empresas a mantener sus prestaciones actualizadas, desde el punto de vista de formatos, accesibilidad y sencillez de operaciones, en particular para webs y servicios de comunidades sociales. Todo ésto ha llevado a que las capacidades de los teléfonos móviles inteligentes sean cada vez mayores y se han posicionado como la tecnología de elección para muchos usuarios, entre ellos, los estudiantes universitarios. Es por eso que el uso de dichos dispositivos debe ser contemplado como una herramienta inevitable en la educación superior.

Este trabajo explora la evolución de los teléfonos inteligentes como una herramienta potencialmente poderosa para facilitar una conexión entre el alumno y la institución académica. Se propone desarrollar una aplicación para la Universidad del Aconcagua, de modo que las actualizaciones, notificaciones y/o noticias emitidas por la Universidad, se reflejen de manera sencilla y casi instantánea en la aplicación móvil, a la cual pueda acceder el alumno desde su dispositivo.

## Índice

1. Introducción .....	8
1.1. Definición del problema.....	8
1.2. Justificación del Proyecto .....	8
1.3. Objetivos .....	10
1.3.1. Objetivo general .....	10
1.3.2. Objetivos específicos .....	10
2. Marco teórico .....	11
2.1. Smartphones .....	12
2.1.1. Introducción .....	12
2.1.2. Evolución histórica .....	14
2.1.3. Los smartphones en Argentina.....	16
2.2. Arquitectura ARM vs X86 .....	18
2.3. Apple .....	20
2.4. Mercado móvil .....	23
2.5. iOS.....	24
2.5.1. Arquitectura .....	24
2.5.2. Sistema .....	25
2.5.3. Estructura y Componentes .....	26
2.5.4. Capa de Núcleo del SO .....	26
2.5.5. Capa de Servicios Básicos .....	27
2.5.6. Framework de Servicios Básicos .....	28
2.5.7. Capa de Medios de Comunicación.....	29
2.5.8. Capa Cocoa Touch .....	32
2.5.9. Versiones de iOS.....	34
2.5.10. iOS SDK .....	40
2.5.11. IOS Developer .....	41
2.5.11.1. Introducción .....	41
2.5.11.2. Proceso de alta en AppStore .....	42
2.5.11.3. Asegurarse que la aplicación está lista para el proceso de aprobación .....	42
2.5.11.4. Lo que se puede esperar durante la revisión de la aplicación .....	43
2.5.11.5. Comprobar el estado de la aplicación en iTunes Connect .....	44
2.5.11.6. Determinar la fecha de disponibilidad de la aplicación .....	47

2.5.11.7. Lineamientos de revisión para publicar en AppStore .....	48
2.6. MVC.....	49
2.6.1. Definición y Estructura .....	49
2.6.2. Uso en iOS .....	50
2.7. WebServices.....	51
2.7.1. Definición.....	51
2.7.2. Utilidad.....	52
2.7.3. Relación con App iOS.....	53
2.8. Seguridad y Autenticación .....	53
2.9. Diseño de Interfaces .....	54
2.9.1. Mockups.....	54
2.10. Mobile Apps y las Universidades .....	55
2.10.1. Universidad de Granada: Granada, España. ....	55
2.10.2. Universidad de Navarra: Navarra, España.....	56
2.10.3. Universidad de Alicante: San Vicente del Raspeig, Alicante, España. ....	57
2.10.4. Universidad de Jaén: Andalucía, España. ....	59
2.10.5. Universidad de Phoenix: Estados Unidos.....	60
2.10.6. Universidad de Michigan: Ann Arbor, Michigan, Estados Unidos.....	61
2.10.7. Universidad de Miami: Miami, Estados Unidos.....	62
2.10.8. Universidad de Harvard: Cambridge, Massachusetts, Estados Unidos. ....	64
2.10.9. Universidad de San Diego: San Diego, California, Estados Unidos. ....	64
2.10.10. Universidad de Bradford: Bradford, Reino Unido.....	66
2.10.11. Universidad de Liverpool: Liverpool, Reino Unido.....	67
2.11. Síntesis .....	69
3. Desarrollo.....	70
3.1. Tecnologías .....	70
3.1.1. Versión de XCode.....	70
3.1.2. Lenguaje PHP .....	71
3.1.3. Servidor Web .....	71
3.1.4. Bases de datos .....	72
3.1.5. Webservices .....	73
3.2. Seguridad y Autenticación .....	75
3.2.1. Configurar SSL en el sitio web.....	75
3.2.2. Costo de SSL.....	77
3.3. iOS Developer.....	77
4. Aplicación .....	78
4.1. Investigación .....	78

4.2.	Requerimientos .....	79
4.3.	Arquitectura del sistema.....	81
4.4.	Modelo de datos .....	83
4.5.	Diagrama de clases.....	85
4.6.	Servicio de notificaciones de Apple.....	86
4.6.1.	Calidad de Servicio (QoS) .....	88
4.6.2.	Arquitectura de seguridad .....	88
4.6.3.	Generación de token.....	89
4.6.4.	Token de confianza (Token Trust).....	90
4.7.	Casos de Uso .....	91
4.7.1.	Diagramas de casos de uso.....	92
4.8.	Diseño de interfaces .....	102
4.8.1.	Screenshots.....	102
4.9.	Opinión personal .....	114
5.	Conclusión.....	115
6.	Fuentes .....	116
7.	Anexos.....	121
7.1.	Anexo A: Apple Review Guidelines.....	121
7.2.	Anexo B: Mockups .....	135

# **1. Introducción**

## **1.1. Definición del problema**

En la Universidad del Aconcagua, no existe aún un sistema, en formato móvil, relacionado con el ámbito académico, ya sea para información, consultas vía web o para trámites administrativos.

La tendencia de la tecnología, en la actualidad, es manejar información desde dispositivos portátiles cada vez más pequeños y prácticos, desde Laptops hasta Netbooks, Tablets y Smartphones. Los usuarios actuales se inclinan por realizar trámites y consultas con herramientas versátiles, sencillas y desde cualquier ubicación, en lugar de depender de una computadora en cierto lugar físico con conexión fija.

## **1.2. Justificación del Proyecto**

El uso de dispositivos móviles en instituciones académicas permite no sólo la facilidad para los alumnos de realizar operaciones universitarias, sino también que el sector administrativo aliviane tareas comunes y evitar consultas innecesarias que enlentecen los procedimientos en sus oficinas físicas.

Además, teniendo en cuenta el avance tecnológico, el uso de móviles para estas tareas, junto con las tablets, podrían reemplazar el uso de computadoras convencionales, tales como las PC de escritorio y las notebooks o laptops, ya sea por su portabilidad y tamaños físicos.

Debido a lo dicho precedentemente, se plantea entonces, realizar un trabajo de desarrollo que permita trabajar en un sistema orientado a celulares inteligentes, dispositivos actuales que trabajan ya con un sistema operativo propio. Específicamente se eligió iOS, el sistema operativo móvil creado por Apple para dispositivos iPhone/iPod/iPad.

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Realizar un sistema desarrollado para plataformas móviles que permita manejar tareas académicas de la Universidad del Aconcagua, y obtener información relacionada con la carrera particular del alumno.

### **1.3.2. Objetivos específicos**

Realizar una aplicación utilizando el lenguaje de Objective-C para dispositivos que utilizan el sistema operativo iOS (iPhone/iPod/iPad) con las siguientes características:

- Permitir informar acerca de clases de consulta de las diferentes materias.
- Permitir informar acerca de las fechas de las mesas de exámenes finales.
- Permitir acceder a la inscripción de las mesas de exámenes finales.
- Contener un sistema de notificaciones para mantener informado al alumno.

## 2. Marco teórico

En la actualidad, el avance de la tecnología móvil ha permitido que el mercado de smartphones ocupe un lugar primordial en el uso diario de los usuarios de internet.

La consultora Cisco Systems<sup>1</sup> ordenó un estudio acerca del uso de *smartphones* (teléfonos inteligentes) a una muestra de jóvenes universitarios y trabajadores de entre 18 y 30 años, así como también a jóvenes profesionales de la tecnología. El estudio se realizó en 18 países: Alemania, Argentina, Australia, Brasil, Canadá, China, Corea del Sur, Estados Unidos, Francia, Holanda, India, Japón, México, Polonia, Reino Unido, Rusia, Sudáfrica y Turquía. El estudio fue realizado a finales de 2012 y puede ser accedido en el sitio de Cisco bajo el nombre de *Cisco Connected World Technology Report*.

Algunas de los valores estadísticos que lanza la investigación, brindados por la empresa InsightExpress<sup>2</sup>, son los siguientes:

- 9 de 10 encuestados globalmente (90%) chequean sus smartphones como parte de la rutina matutina para prepararse para la escuela, universidad o el trabajo.
- 1 de 4 de los encuestados (29%) dice chequear sus smartphones tantas veces que pierde la cuenta.
- Globalmente, 1 de 5 revisa su smartphone para ver correos electrónicos, textos, y actualizaciones de redes sociales al menos cada 10 minutos. En Estados Unidos, 2 de 5 chequean al menos una vez cada 10 minutos.
- Un tercio de los encuestados mira su smartphone al menos una vez cada 30 minutos; en Estados Unidos, la cifra salta a más del 50%.
- El 60% chequea compulsivamente sus smartphones para ver correos electrónicos, textos o actualizaciones de redes sociales.
- De las personas encuestadas, las mujeres se sienten más impulsadas a conectarse: el 85% de las mujeres versus el 63% de los hombres mira compulsivamente sus smartphones para ver textos, correos electrónicos y actualizaciones de redes sociales.

---

<sup>1</sup>*Cisco Systems*: es una empresa dedicada principalmente a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.

<sup>2</sup>*InsightExpress*: es un proveedor de análisis de datos e investigaciones de marketing de la industria, especializada en la medición de marcas de comunicaciones.

- Más del 40% de los encuestados sufriría “abstinencia” y se “sentiría ansioso, como si me faltara una parte” si no pudiera revisar su smartphone constantemente.
- De aquellos usuarios compulsivos de smartphones, el 60% desearía no sentirse tan obligado.
- Globalmente, 3 de 4 encuestados utilizan los smartphones en la cama.
- Más de un tercio utiliza los smartphones en el baño.
- Casi la mitad de los encuestados globales (46%) confesó enviar textos, ver correos electrónicos y chequear redes sociales durante las comidas con familia y amigos. Más de la mitad de los encuestados de Estados Unidos (56%) utiliza smartphones durante comidas sociales.
- Un hábito muy peligroso: 1 de 5 admitió enviar textos mientras maneja.
- El 40% pasa más tiempo con amigos online que con amigos en persona.
- Casi el 90% sube fotos para compartir o almacenar en sitios de internet.
- El 62% sube videos para compartir o almacenar en sitios de internet.
- El 87% tiene una cuenta de Facebook y 1 de 10 tiene siempre el Facebook abierto.
- El 41% actualiza Facebook al menos una vez al día, y 1 de 5 actualiza Facebook varias veces al día.
- El 56% de los encuestados tiene una cuenta de Twitter, y el 21% twitea al menos una vez al día.
- 4 de 5 (81%) cree que las personas tienen distintas identidades online y offline.
- Más de un tercio de los encuestados globales sintió que la mayoría de las personas tienen distintas identidades online versus offline.

## **2.1. Smartphones**

### **2.1.1. Introducción**

Como evolución a los teléfonos celulares convencionales, los *smartphones* o *teléfonos inteligentes* son dispositivos que integran las mismas funcionalidades que un teléfono celular ordinario, añadiendo además funciones características de computadoras personales, como la

multitarea, la posibilidad de instalar aplicaciones de terceros desarrolladores, conectividad inalámbrica WiFi, etc.

Los smartphones son claramente una evolución y fusión entre los teléfonos móviles usuales y las legendarias PDA (Personal Digital Assistant)<sup>3</sup>, compuestos por potentes procesadores y memorias de gran capacidad, generando dispositivos de alto rendimiento y de reducidos tamaños, prácticos y confortables para utilizar y transportar diariamente.

Normalmente los smartphones están compuestos por las siguientes partes:

- *Sistema Operativo (SO)*: es el software que gestiona los recursos del dispositivo y las aplicaciones que corren en él. Existen varios SO para dispositivos móviles en la actualidad, como lo son entre otros iOS, Android, Windows Phone, etc.
- *Conexión Inalámbrica*: además de las redes GSM (Global System for Mobile Communications) utilizadas para las comunicaciones de telefonía móvil, existen otras conexiones que nos permiten el enlace a redes, tanto locales como globales. Podemos destacar como principal la conexión WiFi.
- *GPS (Global Positioning System)*: son sistemas de geo-localización implantados en el dispositivo, que nos permiten determinar la ubicación exacta del mismo. Es útil para identificar rutas desconocidas, localizar dispositivos extraviados y/o robados, informarnos de productos y servicios necesarios a nuestro alrededor, o simplemente avisar de nuestra posición en redes sociales, entre otros usos. No todos los smartphones lo poseen.
- *Cámaras integradas*: son pequeños dispositivos incorporados al celular que permiten capturar imágenes y videos de gran calidad. Los dispositivos más modernos contienen cámaras de captura y grabación en *HD*<sup>4</sup> y en los casos más modernos, hasta en 3D.
- *Aplicaciones nativas*: son las aplicaciones que dan vida y personalizan el dispositivo. Las más fundamentales y básicas (como son la función de llamadas, mensajes, etc.) están elaboradas por el fabricante, pero la gran mayoría son realizadas por terceros, desarrolladores y/o empresas de software.

---

<sup>3</sup>PDA (Personal Digital Assistant): son dispositivos de bolsillo utilizados principalmente como agendas electrónicas, que poseen las características básicas de una computadora.

<sup>4</sup>HD: acrónimo en inglés de High Definition. Contenido multimedia de alta definición.

### 2.1.2. Evolución histórica

Los dispositivos móviles comienzan a principios de la segunda guerra mundial, cuando se hacía fundamental la continua comunicación entre los ejércitos militares.

El primer dispositivo utilizado para estos fines, es el *Handie Talkie H12-16*, creado por *Motorola*, que podría considerarse como el primer teléfono celular, aunque era más una radio para comunicaciones.

Sin embargo, en 1973, de la mano del ingeniero electrónico *Martin Cooper*, quien luego se conocería como el “padre de la telefonía móvil”, se crea el primer teléfono celular comercial, el *Motorola DynaTAC 8000X*, que producido llegaría, en el año 1983, a la mano de los más adinerados de la época. El dispositivo pesaba alrededor de 800 gramos y su batería sólo duraba no más de 1 hora. Sin embargo, como era la tecnología del momento, si se quería tener uno propio, había que pagar aproximadamente unos 4000 dólares.

*"La gente quiere hablar con otras personas, no desde su casa o su oficina, ni desde un coche. Si existe la oportunidad, la gente va a exigir comunicarse libremente en cualquier momento, sin ser impedido por el limitante cable que evita la libertad buscada." (Martin Cooper, 1973)*

Con este simple pensamiento, Cooper originó la evolución en la manera de comunicarse. Esta evolución fue dividida, según su tecnología y servicios, en diferentes etapas llamadas *generaciones*.

La *primera generación (1G)* comprende desde el primitivo terminal creado por Cooper, hasta finales de los años 80', donde se utilizaba una tecnología netamente analógica y estrictamente para voz. La tecnología predominante de esta generación fue la *AMPS* (Advanced Mobile Phone System) creada por *Bell*<sup>5</sup>, la cual se mantuvo funcionando hasta 2008. Esta tecnología usaba frecuencias llamadas “canales” que se asignaban para cada conversación en particular, por lo que requería de un ancho de banda considerable cuando el número de usuarios aumentaba.

---

<sup>5</sup>*Bell*: Empresa canadiense de comunicaciones que ofrece este servicio.

La *segunda generación (2G)*, utilizada en la década de los 90', sustituye la tecnología analógica por la digital. En esta época se empieza a utilizar protocolos de codificación para la comunicación. Para esta generación, se había comenzado a trabajar en un sistema de comunicaciones y bajo la aprobación de implementarlo como un estándar único, se origina el uso de un sistema de telefonía móvil digital a 900 MHz, conocido como *GSM (Global System for Mobile Communications)*. En esta etapa se comienza a utilizar los viejos y conocidos mensajes de texto *SMS (servicio de mensajes cortos)* y *MMS (mensajes multimedia)*, actuando a una velocidad de transferencia de 9 Kbits/s.

Otras tecnologías importantes fueron: *IS-136*, conocida como *TIA/EIA136* o *ANSI-136*; *CDMA (Code Division Multiple Access - Acceso Múltiple por División de Código)* y *PDC (Personal Digital Communications)* que fue utilizado mayormente en Japón.

Entre la segunda y tercera generación, existen 2 generaciones trascendentales que son la *generación 2.5G* y la *generación 2.75G*. La *2.5G* amplía el estándar GSM, originando el servicio *GPRS (General Packet Radio Service – Servicio General de Paquetes vía Radio)*, que trabaja con una transferencia de datos de hasta 114 Kbits/s, aunque el rendimiento real ronda alrededor de los 40 Kbits/s. Con la generación *2.75G* nace el estándar EDGE (Enhanced Data Rates for GSM Evolution – Tasas de Datos Mejoradas para la Evolución de GSM). Esta tecnología trabaja como puente entre la 2G y la 3G, también fue considerada como una evolución de la tecnología GPRS, ya que cuadriplica su rendimiento. Su velocidad de transferencia podría llegar a alrededor de los 470 Kbits/s, aunque solo lo hace a 380 Kbits/s, para respetar las normas de la *ITU*<sup>6</sup> (International Telecommunication Union) y la *IMT-2000*<sup>7</sup>. Esta generación es aplicada a principios del siglo XXI.

La *tercera generación (3G)* maneja las tecnologías de las antiguas generaciones con nuevas tecnologías. Las especificaciones *IMT-2000* de la ITU definieron las características de la 3G. El estándar utilizado en esta generación es el *UMTS (Universal Mobile Telecommunications System - Sistema Universal de Telecomunicaciones Móviles)*. Estas tecnologías permiten la transferencia de video y audio simultáneo y datos a velocidades muy altas, utilizando bandas de 5 MHz que permiten una transferencia de 380 Kbits/s a 2 Mbits/s.

---

<sup>6</sup>*ITU (International Telecommunication Union)*: es el organismo especializado para la información y la comunicación de (TICs).

<sup>7</sup>*IMT-2000 (Telecomunicaciones Móviles Internacionales 2000)*: es el estándar global para la tercera generación de redes de comunicaciones inalámbricas, definidas por la ITU.

Algunos de los servicios importantes que trae la tecnología 3G son el acceso a Internet y las transmisiones de video e imagen en tiempo real, por lo que comienzan a utilizarse en mayor medida las llamadas *videoconferencias*.

A partir del 3G es donde los smartphones toman vital importancia en el mercado, ya que comienzan a ser los dispositivos predilectos para los usuarios y el estilo de vida de los mismos comienza a dar un giro esencial.

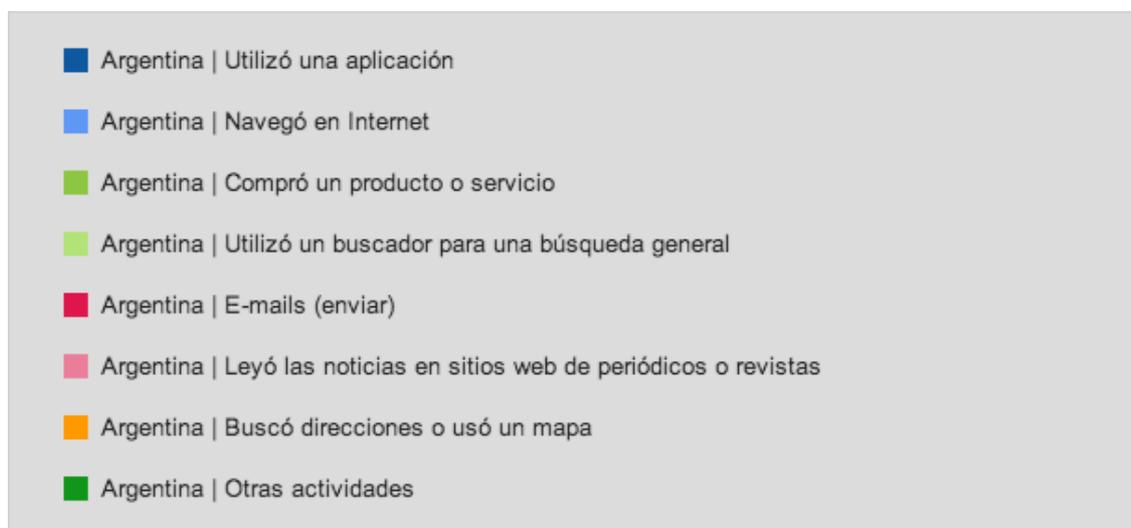
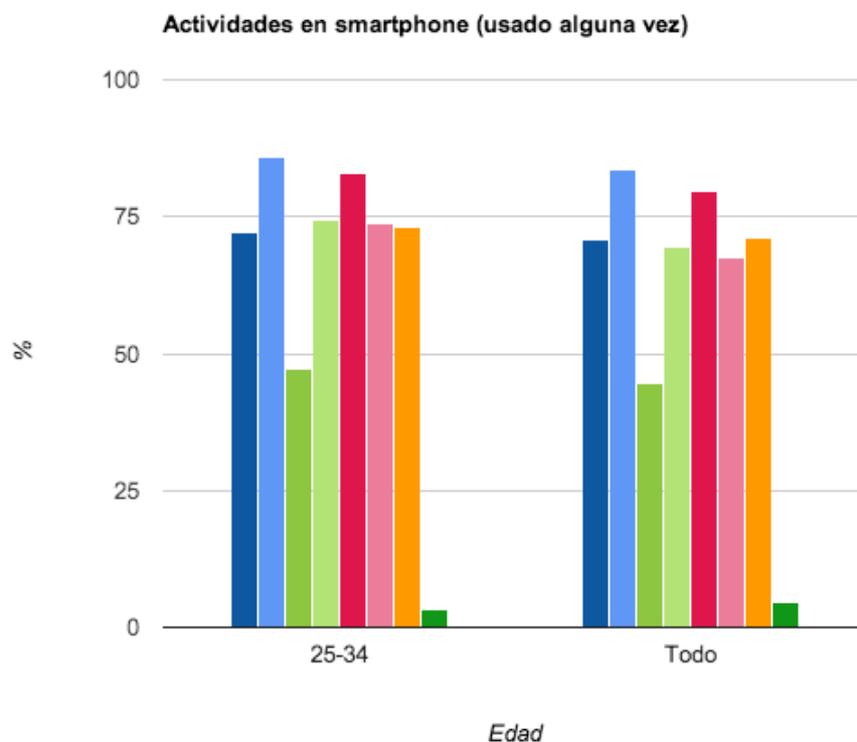
### **2.1.3. Los smartphones en Argentina**

De acuerdo al informe sobre el estudio realizado por Google a través de su portal *Our Mobile Planet*<sup>8</sup>, Mayo 2013, el 31% de la población argentina es poseedora de un smartphone. De los mismos, el siguiente gráfico muestra algunas de las tareas que realizan a diario con sus dispositivos.

*Gráfico N°1: Gráfico sobre actividades realizadas con smartphones en Argentina.*

---

<sup>8</sup>*OurMobilePlanet.com* proporciona acceso a todo el conjunto de datos de este estudio a través de una herramienta de creación de gráficos o la descarga directa de los archivos de datos de un país.



*Fuente: Our Mobile Planet, Google. Fecha: Mayo, 2013.*

Haciendo referencia a los sitios donde usualmente se utilizan los smartphones, el 96% de los usuarios interactúa con los dispositivos en sus domicilios, luego el 88% lo realiza mientras se trasladan a sus destinos, ya sea caminando o en transportes públicos. El 78% utiliza los smartphones en negocios, principalmente el trabajo, sitios de comida, en eventos sociales, en salas de espera, en instituciones educativas, aeropuertos, etc., como se aprecia en el siguiente gráfico:

**Gráfico N°2:** Gráfico sobre lugares de uso frecuente de smartphones en Argentina.



*Fuente:* Our Mobile Planet, Google. *Fecha:* Mayo, 2013.

## 2.2. Arquitectura ARM vs X86

La historia de ARM (Advanced RISC Machine) y su batalla contra X86 viene desde varias décadas, no comenzó con los smartphones. La arquitectura X86 lleva años trabajando en las PC's de escritorio, así como en PC's portátiles como son las notebooks, y parece un recién llegado cuando tratamos de procesadores móviles. Tanto el uno como el otro son conjuntos de instrucciones o ISA (Instruction Set Architecture), se trata de la parte orientada a la programación, el tipo de máquina o traductor que convierte las líneas de código en instrucciones ejecutables. Si se quiere ejecutar un programa diseñado para una de las dos arquitecturas específicamente, se notará que el otro procesador no tiene la capacidad de manejarlo. Esto hace que la ejecución de programas para la arquitectura X86 sobre dispositivos que utilizan ARM sea imposible de procesar.

El tipo de procesador más utilizado en equipos de escritorio es el X86. Este se basa en la arquitectura CISC (Complex Instruction Set Computing), cual brinda un soporte para instrucciones complejas, simultáneas y de ejecución lenta, que a pesar de simplificar la estructura

de programación, termina ganando un alto desempeño, desafortunadamente anclado a un mayor consumo de energía y necesidad de más espacio físico.

Por otro lado, los procesadores ARM son de tipo RISC (Reduced Instruction Set Computer), cuyas propiedades son instrucciones de tamaño fijo con pocos formatos donde sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos. El objetivo de diseñar máquinas con esta arquitectura es facilitar el paralelismo en la ejecución de instrucciones y permitir realizar tareas menores con procesos más cortos, lo que lleva a una disminución importante de la energía utilizada.

Intel y AMD son los mas importantes productores que han hecho evolucionar los procesadores X86 en entornos de escritorio, mientras que el desarrollo de los chips ARM han venido de la mano de distintos fabricantes como Qualcomm, Texas Instruments, Apple, Samsung, entre otros.

Al pensar en el desarrollo de dispositivos que necesitan funcionar y permanecer activos en la totalidad del día, claramente el uso de procesadores ARM es la opción por la que se inclinan las empresas productoras de smartphones... por ahora.

### 2.3. Apple

Apple es una empresa multinacional de Estados Unidos que diseña y produce dispositivos electrónicos y software. Entre los productos de hardware más conocidos de la empresa se encuentran los equipos Macintosh, el iPhone, el iPod y el iPad. Los desarrollos de software más populares de Apple son el sistema operativo Mac OS X, el sistema operativo iOS (para dispositivos móviles como son el iPhone/iPod/iPad), el navegador web Safari y el explorador de contenido multimedia iTunes, siendo este también el portal principal de venta de diferentes productos multimedia, como discos de audio, películas, aplicaciones para sus dispositivos, etc.

BrandZ, la base de datos de marcas más importante del mundo, contiene datos sobre marcas reunidas en varias entrevistas con más de 150.000 personas cada año, siendo la muestra consumidores comunes y profesionales. Todos los años realizan estudios para evaluar las marcas en un contexto competitivo de acuerdo a varios factores, entre ellos el posicionamiento mundial.

Apple, según los reportes de BrandZ, se ubica en el primer puesto compitiendo con otras empresas tecnológicas a nivel mundial, como se observa en la siguiente tabla:

*Tabla N°1: Lista de las 20 marcas tecnológicas más valuadas del periodo 2013.*

## Top 20 Technology

	Brand value 2013 \$M	Brand contribution	Brand value % change 2013 vs 2012
1 <b>Apple</b>	185,071	4	1%
2 <b>Google</b>	113,669	3	5%
3 <b>IBM</b>	112,536	3	-3%
4 <b>Microsoft</b>	69,814	3	-9%
5 <b>SAP</b>	34,365	2	34%
6 <b>Tencent</b>	27,273	4	52%
7 <b>Samsung</b>	21,404	3	51%
8 <b>Facebook</b>	21,261	4	-36%
9 <b>Baidu</b>	20,443	5	-16%
10 <b>Oracle</b>	20,039	2	-11%
11 <b>Accenture</b>	16,503	3	2%
12 <b>HP</b>	16,362	2	-29%
13 <b>Intel</b>	13,757	2	-12%
14 <b>Siemens</b>	12,331	1	16%
15 <b>Cisco</b>	11,816	2	-11%
16 <b>Yahoo!</b>	9,826	3	New
17 <b>Sony</b>	7,786	3	-9%
18 <b>Dell</b>	4,939	2	-25%
19 <b>Philips</b>	4,739	2	New
20 <b>Canon</b>	4,539	2	-33%

Valuations include data from BrandZ™, Kantar Worldpanel, Kantar Retail and Bloomberg.  
Brand Contribution measures the influence of brand alone on earnings, on a scale of 1 to 5 (5 highest).



*Fuente: BrandZ. Fecha: 2013.*

Lo que es aún más admirable sobre la empresa informática, es que dentro de los registros ocupa también el primer puesto en empresas a lo que refiere a marcas en general, es decir, en la mayoría de las categorías. La siguiente tabla muestra su posicionamiento respecto a otras.

Tabla N°2: Lista de las 100 marcas mas valuadas del periodo 2013.

### BRANDZ™ Top 100 Most Valuable Global Brands 2013

Category	Brand	Brand value 2013 \$M	Brand contribution	Brand value % change 2013 vs 2012	Rank change
1 Technology	Apple	185,071	4	1%	0
2 Technology	Google	113,669	3	5%	1
3 Technology	IBM	112,536	3	-3%	-1
4 Fast Food	McDonald's	90,256	4	-5%	0
5 Soft Drinks	Coca-Cola	78,415	5	6%	1
6 Telecoms	AT&T	75,527	3	10%	2
7 Technology	Microsoft	69,814	3	-9%	-2
8 Tobacco	Marlboro	69,383	3	-6%	-1
9 Credit Card	VISA	56,080	4	46%	6
10 Telecoms	Verizon	55,568	3	18%	0
11 Conglomerate	General Electric	55,357	2	21%	0
12 Telecoms	Verizon	53,004	3	8%	-3
13 Regional Banks	Bank of America	47,748	3	20%	1
14 Retail	amazon.com	45,727	3	34%	4
15 Logistics	UPS	42,747	5	15%	1
16 Regional Banks	Wells Fargo	41,115	2	-1%	-3
17 Telecoms	Verizon	38,712	3	-8%	-5
18 Retail	Walmart	38,220	2	5%	-1
19 Technology	SAP	34,365	2	34%	3
20 Credit Card	MasterCard	27,821	4	34%	9
21 Technology	Tencent 腾讯	27,273	4	62%	16
22 Regional Banks	China Construction Bank	26,859	2	10%	2
23 Cars	BMW	24,497	4	12%	5
24 Cars	Mercedes-Benz	24,015	4	-2%	-1
25 Global Banks	HSBC	23,970	3	24%	6

Values include data from BrandZ™. Source: Millward Brown Retail and Branding. Brand Contribution measures the influence of brand name on earnings, on a scale of 1 to 5 (highest).



Category	Brand	Brand value 2013 \$M	Brand contribution	Brand value % change 2013 vs 2012	Rank change
26 Entertainment	Disney	23,913	3	49%	17
27 Telecoms	China Mobile	23,893	2	-11%	-7
28 Credit Card	Amex	23,514	4	16%	2
29 Luxury	Louis Vuitton	22,719	4	-12%	-8
30 Technology	LinkedIn	21,404	3	51%	25
31 Technology	Facebook	21,261	4	-36%	-12
32 Baby Care	Pampers	20,594	5	13%	3
33 Technology	Baidu	20,443	5	-16%	-8
34 Beer	Heineken	20,297	4	28%	14
35 Apparel	ZARA	20,167	3	60%	31
36 Technology	ORACLE	20,039	2	-11%	-9
37 Regional Banks	中国农业银行	19,975	2	12%	1
38 Regional Banks	Bank of China	19,968	4	16%	2
39 Oil & Gas	ExxonMobil	19,229	1	5%	-6
40 Luxury	Cartier	19,129	4	0%	-8
41 Retail	Walmart	18,488	2	43%	21
42 Personal Care	L'OREAL	17,971	4	30%	15
43 Cars	Mercedes-Benz	17,952	4	11%	3
44 Fast Food	Starbucks	17,892	4	5%	-2
45 Personal Care	Gillette	17,823	4	-6%	-12
46 Regional Banks	TD	17,781	4	22%	8
47 Retail	ebay	17,749	2	40%	17
48 Regional Banks	Commonwealth Bank	17,745	3	36%	12
49 Oil & Gas	Shell	17,678	1	-1%	-10
50 Personal Care	Colgate	17,250	4	15%	1

The Brand Value of Chinese Brands (CNY), New and Old. The Brand Value of Business includes B2B CNY.

### BRANDZ™ Top 100 Most Valuable Global Brands 2013

Category	Brand	Brand value 2013 \$M	Brand contribution	Brand value % change 2013 vs 2012	Rank change
51 Fast Food	McDonald's	16,591	4	12%	1
52 Regional Banks	Bank of America	16,565	3	New	New
53 Technology	accenture	16,503	3	2%	-8
54 Technology	hp	16,302	2	-29%	-28
55 Retail	TESCO	16,300	4	-9%	-19
56 Apparel	NIKE	15,817	4	-3%	-12
57 Insurance	中国人寿	15,279	3	5%	-4
58 Regional Banks	中国工商银行	14,236	2	10%	3
59 Regional Banks	ICICI Bank	14,196	1	12%	4
60 Telecoms	中国移动	13,829	2	-10%	-10
61 Technology	intel	13,757	2	-12%	-12
62 Logistics	FedEx	13,732	4	17%	8
63 Regional Banks	us bank	13,716	3	19%	9
64 Global Banks	citi	13,396	2	37%	18
65 Oil & Gas	PetroChina	13,380	1	11%	3
66 Telecoms	movistar	13,336	2	-22%	-25
67 Oil & Gas	BP	13,127	1	-6%	-11
68 Luxury	GUCCI	12,735	5	48%	New
69 Apparel	H&M	12,732	2	-6%	-11
70 Regional Banks	SBERBANK	12,655	3	19%	4
71 Cars	Volvo	12,401	3	-2%	-6
72 Technology	SIEMENS	12,331	1	16%	1
73 Alcohol	Heineken	12,193	3	3%	-4
74 Retail	IKEA	12,040	3	31%	15
75 Soft Drinks	pepsi	12,029	4	-5%	-8

Values include data from BrandZ™. Source: Millward Brown Retail and Branding. Brand Contribution measures the influence of brand name on earnings, on a scale of 1 to 5 (highest).



Category	Brand	Brand value 2013 \$M	Brand contribution	Brand value % change 2013 vs 2012	Rank change
76 Retail	Walmart	11,879	3	13%	0
77 Technology	cisco	11,816	2	-11%	-18
78 Oil & Gas	BP	11,520	1	11%	-1
79 Telecoms	MTN	11,448	3	23%	9
80 Retail	Woolworths	11,039	3	New	New
81 Global Banks	CHASE	10,836	3	25%	11
82 Telecoms	AT&T	10,633	3	11%	3
83 Soft Drinks	Red Bull	10,558	3	6%	-3
84 Insurance	中国平安	10,558	3	4%	-6
85 Regional Banks	Santander	10,396	2	8%	-2
86 Cars	Subaru	10,186	3	3%	-5
87 Global Banks	Standard Chartered	10,160	2	1%	-8
88 Regional Banks	Wingtip Australia's First Bank	10,070	3	New	New
89 Telecoms	telcel	10,054	3	-13%	-18
90 Telecoms	docomo	10,028	3	-37%	-43
91 Fast Food	McDonald's	9,953	3	12%	0
92 Technology	YAHOO!	9,826	3	New	New
93 Global Banks	J.P.Morgan	9,668	2	New	New
94 Telecoms	BT	9,521	2	New	New
95 Luxury	PRADA	9,454	4	63%	New
96 Global Banks	Santander	9,232	3	8%	-1
97 Oil & Gas	Shell	9,036	1	5%	-4
98 Logistics	DHL	8,940	3	18%	2
99 Retail	ASDA	8,885	2	-5%	-12
100 Cars	VW	8,790	3	3%	-4

The 2013 Brand Value of Spain has been included in the 2013 Brand Value of Europe. The Brand Value of Russia includes B2B CNY. The Brand Value of Russia includes B2B CNY.

Fuente: BrandZ. Fecha: 2013.

Como muestran las imágenes, la empresa Apple está posicionada en el primera ubicación de marcas a nivel mundial, valuada en alrededor de 185 millones de dólares.

Debido a que los productos de Apple tienen un fuerte impacto en mercado, y teniendo en cuenta los millones de usuarios propietarios de estos dispositivos, se eligió la plataforma de iOS para desarrollar una aplicación móvil disponible para celulares iPhone y dispositivos móviles de Apple (iPod e iPad).

## **2.4. Mercado móvil**

Apple lanzó la primer tienda de aplicaciones móviles, la cual revolucionó el mercado de la telefonía móvil. Igualmente, debido a la importante demanda de aplicaciones en diferentes plataformas, las tiendas han crecido estos últimos años, tanto en tecnología, control y seguridad, como también en sus sistemas de compra y cobro, permitiendo confiar cada vez más en ellas, y permitiendo abarcar un mercado mayor con una tasa mínima de inversión.

## 2.5. iOS

### 2.5.1. Arquitectura

La arquitectura de iOS es similar a la que se utiliza en Mac OS X. En el nivel más alto, iOS actúa como un intermediario entre el hardware del dispositivo y las aplicaciones que aparecen en su pantalla.

Las aplicaciones no se comunican con el hardware directamente, sino que lo hacen a través de un conjunto de interfaces de sistema definidas, que protegen la aplicación de los cambios de hardware. Esta abstracción facilita la creación de aplicaciones para que sean capaces de funcionar en dispositivos con diferentes características de hardware.

La imagen a continuación muestra las diferentes capas que forman la arquitectura descrita anteriormente.

*Imagen N°1: Aplicaciones iOS mostradas en capas.*



*Fuente: Elaboración propia basado en iOS Technology Overview, Apple. Fecha: 2012*

Cuando se escribe un código o se programa para iOS, se debe tratar de hacer sobre los frameworks de alto nivel, es decir, elegirlos antes que los de bajo nivel, siempre que sea posible. Los frameworks de alto nivel son los que proporcionan las abstracciones orientadas a objetos que permiten las de bajo nivel. Estas abstracciones hacen que sea mucho más sencillo escribir código, ya que reduce la cantidad de código que se tiene que escribir y encapsula las características que son más complejas, como son los sockets y los threads, por ejemplo.

### **2.5.2. Sistema**

El sistema iOS ofrece un conjunto de interfaces para acceder a las diversas características de bajo nivel del SO. Las interfaces están basadas en el lenguaje de programación C y proveen soporte a:

- Hilos (Threads)
- Red (Sockets)
- Acceso al sistema de archivos
- Estándares de E/S
- Servicios DNS
- Asignación de Memoria
- Cálculos matemáticos

### 2.5.3. Estructura y Componentes

La implementación de las tecnologías de iOS se pueden ver como un conjunto de capas, donde las capas más bajas del sistema poseen los servicios fundamentales y las tecnologías que precisan todas las aplicaciones, mientras que en las capas más altas se encuentran los servicios y tecnologías más sofisticados. La siguiente imagen muestra las capas que contiene el sistema:

*Imagen N°2: Capas de la arquitectura de iOS.*



*Fuente: Elaboración propia basado en iOS Technology Overview, Apple. Fecha: 2012.*

### 2.5.4. Capa de Núcleo del SO

El núcleo del Sistema Operativo (SO) contiene las mismas características que cualquier otro SO en sus capas de bajo nivel. Al igual que otros sistemas, para utilizar estas tecnologías en una aplicación, se hace uso de diferentes frameworks. Estos permiten la comunicación con el hardware externo. Algunos frameworks son:

- *Framework acelerador (Accelerate Framework):* contiene funciones *DSP*<sup>9</sup>(procesador de señales digitales) y acelerador para procesos de cálculos (matemáticos).

<sup>9</sup> Las funciones DSP permiten la comunicación a través de las redes inalámbricas y señales de comunicaciones (WiFi, GSM, CDMA, WiMAX, etc).

- *Framework de Bluetooth (Core Bluetooth)*: permite al desarrollador interactuar con las funcionalidades y accesorios que utilizan la tecnología Bluetooth.
- *Framework de accesorios externos (External Accessory Framework)*: provee el soporte para la comunicación con hardware externo que se conecte al dispositivo basado en iOS.
- *Framework de Seguridad (Security Framework)*: se utiliza para garantizar la seguridad de los datos que administran las aplicaciones. Proporciona interfaces para la gestión de certificados, claves públicas y privadas y políticas de confianza.

### 2.5.5. Capa de Servicios Básicos

Esta capa contiene los servicios de sistema fundamentales que utilizan todas las aplicaciones. Si bien no son utilizados directamente, muchas partes del sistema están construidos sobre ellos.

Algunas de las principales tecnologías que se encuentran disponibles en la capa de servicios básicos pueden ser:

- *iCloud*: nace con iOS 5 y permite, entre otras cosas, crear, ver, modificar y transferir documentos desde cualquier dispositivo Apple sincronizado a la cuenta iCloud del usuario.
- *In-App Purchase*: aparece en iOS 3, esta tecnología brinda la posibilidad de vender productos y servicios dentro de una aplicación nativa, usando la cuenta de usuario de iTunes.
- *SQLite*: la librería SQLite permite embeber una sencilla (pero poderosa) base de datos SQL en la aplicación iOS sin necesidad de utilizar una base de datos desde un servidor remoto. Esto permite un manejo local de información, más ágil y sin necesidad de conexión a internet.
- *Soporte XML*: la *Foundation Framework*<sup>10</sup> proporciona la clase `NSXMLParser` para utilizar elementos de un archivo XML. Esta clase permite manipular contenidos XML

---

<sup>10</sup>The *Foundation Framework* define una capa base de clases para Objective-C. Proporciona un conjunto de clases primitivas e introduce varios paradigmas que definen las funcionalidades no cubiertas en el lenguaje Objective-C.

proporcionados por la biblioteca libXML2, una biblioteca open source que analiza y escribe datos XML de forma rápida y transforma su contenido a HTML.

### 2.5.6. Framework de Servicios Básicos

Algunos de los frameworks pertenecientes a la capa de servicios básicos son:

- *Framework de cuentas (Accounts.framework)*: brinda un modelo de inicio de sesión para ciertas cuentas de usuario. Mejora la experiencia del usuario, ya que no tienen que solicitar al mismo datos por separado para registros relacionados con una cuenta. Un ejemplo común, es la cuenta de Twitter, donde trabaja en conjunto con el framework de Twitter para acceder a una cuenta personal.
- *Framework de Agenda (AddressBook.framework)*: permite el acceso a los contactos almacenados en el dispositivo. Si la aplicación utiliza información de un contacto, se puede utilizar este framework para accederlo e incluso modificarlo en la base de datos.
- *Framework de Red (CFNetwork.framework)*: contiene soluciones para trabajar con protocolos de red. Estas abstracciones permiten utilizar de manera sencilla los niveles inferiores, tales como los sockets. Se utilizan para simplificar las comunicaciones vía FTP y HTTP.
- *Framework de modelo de datos (CoreData.framework)*: es la tecnología que administra el modelo de MVC (Modelo-Vista-Controlador). En lugar de programar un modelo de datos para realizar la aplicación, se pueden utilizar las herramientas gráficas de XCode para construir un esquema que represente el modelo de datos.
- *Framework de Localización (CoreLocation.framework)*: provee servicios de localización. Utiliza el hardware de GPS o radios WiFi para encontrar la posición actual del dispositivo y/o usuario, con latitud y longitud.
- *Framework de Servicios de Teléfono (CoreTelephony.framework)*: proporciona interfaces para interactuar con los servicios de teléfono en dispositivos que lo soportan. También toma información sobre el proveedor de servicios de telefonía celular.

- *Newsstand Kit Framework*: introducido en iOS 5, provee un lugar común para leer y guardar revistas y diarios. Los editores que desean ofrecer sus publicaciones a través de una aplicación iOS, deben hacerlo utilizando esta herramienta.

### **2.5.7. Capa de Medios de Comunicación**

Podría denominarse como la “capa multimedia”, la cual contiene las características de audio, video, gráficos y tecnologías orientadas a crear una mejor experiencia multimedia para el usuario. Estas tecnologías también hacen que sea fácil crear aplicaciones, con buen sonido y gran aspecto visual.

#### **2.5.7.1. Tecnología Gráfica:**

Una parte importante de todas las aplicaciones iOS es su alta definición en gráficos. Las tecnologías que administran los contenidos gráficos de las aplicaciones son:

- *Quartz*, encargado de renderizar vectores e imágenes 2D.
- *CoreAnimation*, parte de *Quartz*, soporte para vistas animadas y otros contenidos.
- *CoreImage*, prestador de soporte par manipulación de video e imágenes fijas.
- *OpenGL* y *GLKit*, soporte para render 2D y 3D utilizando acelerador de hardware.
- *CoreText*, diseño sofisticado de texto y motor de renderizado.
- *Imágenes de E/S*, que proporciona interfaces para leer y escribir la mayoría de los formatos de imagen.

Para las aplicaciones que trabajan en dispositivos con *Retina Display*<sup>11</sup> deben realizarse algunas modificaciones, ya que el contenido se redimensiona automáticamente en los dispositivos de alta resolución para mejorar su visualidad.

### 2.5.7.2. Tecnología de Audio:

Las tecnologías de audio de iOS están diseñadas para proporcionar al usuario una experiencia de audio de gran calidad, tanto para reproducción como para grabación.

El sistema ofrece varias formas de reproducir y grabar contenidos de audio, gracias al *Media Player Framework*. Este framework provee el acceso de manera sencilla a la librería de iTunes para reproducir música y listas de audio.

En conjunto con el anterior, el *AV Foundation Framework* proporciona un conjunto de interfaces de Objective-C de fácil uso para administrar la reproducción y grabación.

También ofrece *OpenAL*<sup>12</sup>, que proporciona interfaces para la entrega de audio posicional (sonido envolvente).

Los frameworks de audio ofrecen interfaces simples y sofisticadas para reproducir y grabar sonido, así como también para manejar los sonidos de alerta del sistema (en conjunto con las funciones de vibración) y la administración de almacenamiento temporal y reproducción de sonidos para canales locales o de streaming.

---

<sup>11</sup>*Retina Display* es una tecnología de Apple que permite, a través de la reducción de píxeles a lo alto y ancho de la pantalla de los dispositivos (iPhone, iPod y recientemente iPad), visualizar con mayor nitidez texto e imagen.

<sup>12</sup>*OpenAL* es una API multiplataforma de audio en 3D usada para aplicaciones de audio y juegos.

### 2.5.7.3. Tecnología de Video:

iOS ofrece varias tecnologías para reproducir videos o transmitirlos en la red. En los dispositivos que poseen hardware adicional de video (cámara incorporada) se permiten aplicar funciones para capturar video y utilizarlo en la aplicación.

El sistema ofrece varias formas de reproducir y grabar, dependiendo las necesidades particulares del usuario y/o la aplicación. El *Media Player Framework*, que ofrece interfaces de alto nivel, permite implementar las características particulares que se precisan en la aplicación de manera sencilla.

Algunos recursos disponibles para utilizar las tecnologías de video son:

- La clase *UIImagePickerController* que provee interfaces de fácil uso para grabar video en los dispositivos que poseen el hardware adecuado.
- El *Media Player Framework* facilita interfaces para reproducir videos en pantalla completa o en un determinado tamaño.
- El *AV Foundation Framework* permite, mediante un conjunto de interfaces, administrar la reproducción y captura de videos.

Los formatos de video soportados por iOS para video son:

- .mov
- .mp4
- .m4v
- .3gp

## 2.5.8. Capa Cocoa Touch

Esta capa probablemente sea la capa con la que el desarrollador iOS se familiarice más. La capa *Cocoa Touch* contiene los frameworks esenciales para la construcción de aplicaciones iOS. Esta capa define la infraestructura de las aplicaciones y el soporte para tecnologías fundamentales como la multitarea, funciones touch, notificaciones push y otros servicios de alto nivel del sistema.

Algunas de las principales tecnologías disponibles en la capa Cocoa Touch se describen a continuación.

### 2.5.8.1. Storyboards:

Son utilizados para diseñar las interfaces de la aplicación. Los storyboards permiten unificar la construcción de las vistas en un solo lugar y verificar como funcionan en conjunto. Pueden definirse transiciones entre vistas sin necesidad de programarlas en el editor.

Un storyboard puede almacenar todos los controladores de vista de la aplicación, o se pueden usar varios para organizar diferentes partes y funcionalidades dentro de la misma.

### **2.5.8.2. *Multitarea***

A partir de iOS 4, se comienza a implementar el soporte multitarea en los dispositivos iPhone y los recientes (en esa época) iPad. Las aplicaciones no finalizan al presionar el botón inicio, sino que continúan en ejecución, en un segundo plano.

Para conservar batería en el dispositivo, la mayoría de las aplicaciones se suspenden por el sistema cuando entran en background. Estas permanecen en suspensión, es decir, siguen en memoria pero sin ejecutar procesos. De esta manera pueden reanudarse mas rápidamente.

Existen excepciones en las cuales una aplicación puede seguir ejecutándose normalmente en segundo plano y puede ser por las siguientes causas:

- Necesita tiempo para terminar una tarea importante.
- Puede ser imprescindible para otros servicios actuales.
- Puede utilizar notificaciones locales para generar alertas de usuario en un determinado tiempo.

### **2.5.8.3. *Sistema de notificaciones Push***

Desde iOS 3, el sistema implementa una forma de aviso a los usuarios de la información nueva, aun cuando la aplicación no está en ejecución. Este servicio permite enviar notificaciones de texto, notificaciones visuales utilizando adornos en el ícono de la aplicación, o alertas de sonido para enterar al usuario en cualquier momento. Estas notificaciones permiten al usuario saber que debe abrir la aplicación para recibir la información especificada.

Hay que tener en cuenta dos cosas para realizar notificaciones push en una aplicación: primero, la misma debe solicitar la entrega de notificaciones y procesarlas cuando las recibe y;

segundo, se necesita proveer un servicio desde el servidor para generar las notificaciones. Este proceso se realiza desde allí y requiere de *Apple Push Notification Service*<sup>13</sup> para funcionar.

### **2.5.9. Versiones de iOS**

Desde hace ya 5 años, cuando se introdujo el primer iPhone, el sistema móvil de Apple ha sufrido innumerables cambios y mejoras que lo ubican en el puesto más alto de sistemas para móviles de la actualidad.

iOS avanza como lo ha hecho desde Enero de 2007, y los cambios han sido vertiginosos, que a veces pensamos en el primer teléfono y no podemos creer que no tenga las funcionalidades que posee en la actualidad sino ha transcurrido tanto tiempo. A continuación se hará una descripción de las versiones por las que se ha desarrollado iOS.

#### ***2.5.9.1. iOS 1: el inicio del iPhone***

Apple construyó el sistema del iPhone basado en el núcleo de UNIX, usando la mayoría de las herramientas con las se creó el sistema operativo de las computadoras, el Mac OS X. Como estos comparten ciertas similitudes, la empresa pensó en colocarle el mismo nombre (OS X), pero luego decidieron que se trataba de un dispositivo con las características suficientes para justificar su propia marca, por lo que lanzaron al sistema operativo en su móvil como *iPhone OS*.

---

<sup>13</sup> APNs (abreviado) es un servicio robusto y altamente eficiente para la propagación de información a dispositivos como iPhone/iPod e iPad.

Cuando el iPhone es lanzado, el sistema operativo del mismo es nombrado *iPhone OS*, y de hecho mantiene ese nombre hasta 2010, cuando aparece la versión iOS 4. El iPhone OS cambia su nombre por iOS por una cuestión de simplicidad.

En los comienzos del iPhone OS, sus características no diferían demasiado de la competencia y habían muchas restricciones, que a esa época no eran perceptibles: el iPhone no tenía soporte 3G, no tenía multitarea, permitía manejar la administración de correos electrónicos, pero no contemplaba el envío de archivos adjuntos, no permitía visualizar el sistema de archivos al usuario, no tenía la pantalla personalizable, no soportaba edición de documentos de *Office* y no soportaba aplicaciones de terceros, es decir, fue bloqueado casi en su totalidad para los desarrolladores.

Las principales características del nuevo smartphone fueron:

- *Nueva interfaz y manejo de pantalla:* hasta el momento, los teléfonos táctiles más avanzados utilizaban lápices para manejar sus pantallas resistivas. El iPhone cambia esa tecnología por una pantalla capacitiva, permitiendo una interfaz más simple y potente.
- *Sincronización con iTunes:* la fiabilidad que posee iTunes, permitió generar cierta invisibilidad de procesos dispositivo-PC de manera que resultara sumamente sencillo para los usuarios sincronizar sus dispositivos con la computadora.
- *Potente navegador web:* iPhone incluyó el navegador que utiliza todo sistema Mac: *Safari*. La manera de conectarse y navegar por Internet cambió de una manera vertiginosa, conservando casi todas las características que podían apreciarse en una computadora de escritorio, dejando de lado el utilizado lenguaje *WML*<sup>14</sup> para dispositivos móviles.
- *Teclado en pantalla:* la manera con la que se implementó el teclado fue diferente a la competencia, ya que se trató de un teclado inteligente que sólo aparecía cuando se lo necesitaba. Esto permitió mayor espacio en la pantalla para visualizar lo necesario. La manera de utilizarlo cambió de los lápices, que hasta entonces utilizaban los smartphones, a simplemente los dedos.
- *Multitarea sólo para iPod:* el sistema multitarea aún no había sido implementado mas que para reproducir música en modo background. De esta manera se podía escuchar música mientras se realizaba otras tareas.
- *Springboard:* el diseño simple del iPhone hizo que el smartphone constara de un único botón físico: *springboard* consistió en saltar de cualquier lugar donde se encontrara el usuario dentro de iOS hacia lapantalla de inicio. Fue la manera de finalizar cualquier

---

<sup>14</sup> WML es una versión reducida del lenguaje HTML utilizada para visualizar contenido web en dispositivos móviles que trabajan con tecnología WAP.

aplicación que se estuviera ejecutando (con excepción de iPod, como se mencionó anteriormente).

### ***2.5.9.2. iOS 2: aplicaciones***

Luego de aplicar algunas actualizaciones al iOS 1, tales como permitir reorganizar iconos en la pantalla principal, agregar íconos de accesos directos a direcciones web en la misma, etc., y después de la salida del iPod touch entre otros, se introduce el mercado conocido como *AppStore*. La *AppStore* es una tienda de aplicaciones creadas por desarrolladores ajenos a la empresa creadora, lo que permitía generar un nuevo mercado en el ámbito de las tecnologías móviles.

Apple aprovecha la misma base de la cuenta de iTunes que el usuario utiliza para comprar música, de manera que el mismo no debe crear nuevas cuentas con datos personales ni datos de la cuenta bancaria para comprar aplicaciones.

Cuando aparece la *AppStore*, y en incentivo de generar nuevas aplicaciones, se hace necesario para el programador ser provisto de las herramientas para realizarlas. Por esto, Apple lanza el *iPhone SDK*, o lo que hoy se conoce como *iOS SDK*. El *iOS SDK* es el kit de desarrollo para desarrolladores, con las herramientas necesarias para crear potentes aplicaciones para los dispositivos (iPhone/iPod touch y más tarde iPad). En el apartado 3.4 se explica con más detalles.

### ***2.5.9.3. iOS 3: funciones***

El iOS 3 se lanza en conjunto con el iPhone 3G. Este nuevo sistema operativo trae algunas características nuevas como:

- *Cortar, copiar y pegar*: Apple introduce una nueva funcionalidad que permite la selección de texto, cortar, copiar y pegar el mismo en otro sector. Esta funcionalidad hasta el momento sólo se veía en computadoras ordinarias. Los usuarios querían esta funcionalidad desde hacía tiempo, pero Apple se tomó el tiempo de realizar un buen trabajo, respetando sus políticas y estándares.
- *Spotlight*: ya que el proceso de encontrar contenido en iOS resultaba algo tedioso y lento, se implementa el sistema de búsqueda utilizado en MacOS X llamado *Spotlight*. Esta función permite, mediante índices virtuales que poseen todos los archivos del sistema, buscar rápidamente y devolver coincidencias de la búsqueda, sean archivos, aplicaciones, webs, etc.
- *Notificaciones Push*: el propósito de las notificaciones *Push* es habilitar una aplicación para informar a sus usuarios que hay algo pendiente para ellos, ya sea un mensaje, un recordatorio, etc., cuando la aplicación principal no se está ejecutando en primer plano.

Algunas otras características que se aplicaron a iOS 3, en conjunto con funcionalidades de otras aplicaciones fueron:

- Posibilidad de realizar foco tocando el sector de la pantalla, con la aplicación de la cámara.
- Comprar música, libros y películas en iTunes desde el dispositivo.
- Se aplica el control parental para aplicaciones que no son apropiadas para menores.
- Aplicación de brújula (compass).

En la actualización 3.2 del iOS, Apple lanza el nuevo y revolucionario dispositivo llamado *iPad*. Éste era muy parecido en funcionalidades a sus dispositivos compañeros, con la diferencia que poseía una pantalla más grande. Muchas actualizaciones del iOS 3.2 hacen adaptar las características existentes a estas nuevas dimensiones. Se añadieron ventanas emergentes (popups) en reemplazo de las nuevas ventanas que se experimentaban en el iPhone, lo que permitía permanecer en la misma vista en lugar de pasar a una nueva. Junto con esta incorporación del nuevo dispositivo, se hace necesario comenzar con nuevos desarrollos por parte de los programadores.

#### 2.5.9.4. iOS 4: multitarea

Con iOS 4, aparece la característica que se ha aplicado con el tiempo en la mayoría de los sistemas operativos: la *multitarea*.

La función de la multitarea es realizar varios procesos utilizando un mismo procesador para ejecutarlos, es decir que es este contexto nos permite correr varias aplicaciones en un dispositivo al mismo tiempo, por lo que se puede, por ejemplo, copiar algún texto del navegador, abrir el correo y enviar lo citado, luego continuar en el navegador sin éste haber sido cerrado.

Aunque era la principal novedad en los dispositivos iOS, la función de multitarea no lo era del todo, ya que no soporta realmente a todas las aplicaciones para seguir funcionando en segundo plano. También acarreó un detalle importante con respecto a la ejecución de varios procesos, ya sean principales o en segundo plano, y fue el tema de la administración de batería, por lo que las revisiones de aplicaciones se hacen más rigurosas con respecto al ahorro de energía.

- *Facetime*: iOS 4 se lanza junto con el nuevo dispositivo: el iPhone 4. Este añade, entre otras características, la cámara frontal del teléfono. Esto permitía realizar videollamadas, por lo que se implementa la aplicación Facetime. La app es muy similar a Skype, donde se guardan contactos y se pueden realizar llamadas hacia ellos como si se tratase de una llamada convencional, pero agregando video.
- *Carpetas*: a partir de iOS 4, se permite agrupar aplicaciones en carpetas, una solución que permite tener ordenado el escritorio en el dispositivo. Con esta función, el usuario puede organizar y reducir, junto con Spotlight, el tiempo de búsqueda de una aplicación en el teléfono o tableta.
- *Retina Display*: el iPhone 4 posee una mejora significativa con su pantalla, la cual duplica la resolución que traía en sus anteriores dispositivos. Esta mejora permitió a los desarrolladores a crear aplicaciones de mayor calidad y de diseño más nítido, agradable a la vista, aunque también obligó a los mismos a apresurarse en dar soporte a esta nueva resolución.

### 2.5.9.5. iOS 5: SIRI

iOS 5 vino de la mano del iPhone 4S, como fue el caso del iOS 3. También trajo varias nuevas características que permitían al iPhone competir con los smartphones actuales:

- *Siri*: El nuevo dispositivo dispone de una tecnología innovadora, llamada *Siri* que reemplaza el control de voz con un asistente virtual que permite realizar preguntas en el lenguaje natural (humano) y, enlazado con todo el sistema y la web, responde en consecuencia al “deseo” del usuario. Siri ha sido lanzada en su versión Beta y tiene sus limitaciones respecto a los alcances, pero es lo suficientemente inteligente como para responder acerca de estado de tráfico por ejemplo, realizar lectura y envío de mensajes y correos, responder ante avisos y recordatorios, entre otras cosas. Su tecnología sólo está disponible en algunos idiomas como inglés, francés y alemán.
- *Centro de Notificaciones*: debido al disperso sistema de notificaciones que iOS manejaba, se plantea un sistema centralizado similar al utilizado por la competencia *Android*<sup>15</sup>, que concentra todos los avisos de las aplicaciones, junto con otras notificaciones adicionales, como recordatorios, alarmas, etc., en un solo lugar, representada en una vista desplegable de la parte superior del dispositivo. Este sistema reemplaza de alguna manera los molestos popup que eran utilizados por algunas aplicaciones para realizar avisos.
- *Mensajes*: iOS 5 implementa un sistema de mensajería semejante al que utiliza *Blackberry*<sup>16</sup>, otro competidor, donde los usuarios con dispositivos iOS pueden intercambiar mensajes entre ellos, tanto de texto sencillo como mensajes multimedia. Este sistema no solo genera una diferencia entre otros smartphones, sino que también amenaza a las empresas prestadoras de servicios de comunicaciones, ya que este tipo de servicios son cobrados a sus usuarios, mientras que iOS lo ofrece de manera gratuita.
- *iCloud*: con iOS 5 es lanzado un servicio de almacenamiento en la nube que permite sincronizar de manera automática todos los dispositivos personales del usuario, tanto iOS como OS X, de manera que se permita acceder a los contenidos desde cualquier dispositivo. Este servicio ya era conocido con *Dropbox*<sup>17</sup>, ya que tiene un funcionamiento similar, aunque éste permite expandirse a otros dispositivos como Android, BlackBerry, Windows, Linux, etc.

---

<sup>15</sup>*Android* es un Sistema Operativo para dispositivos móviles, creado por la empresa Google.

<sup>16</sup>*BlackBerry* (BB abreviado) es empresa creadora de smartphones que utiliza el Sistema Operativo *RIM* (Research in Motion) para el manejo de estos dispositivos.

<sup>17</sup>*Dropbox* es un servicio que permite compartir archivos en cualquier lugar con facilidad.

### 2.5.10. iOS SDK

El iOS SDK (iOS Software Development Kit) es un paquete de herramientas pensada para el desarrollador, que permite al mismo generar aplicaciones nativas para los dispositivos móviles de Apple: iPhone, iPod e iPad. Estas aplicaciones son escritas en el lenguaje *Objective-C*<sup>18</sup>. El conjunto de utilitarios de desarrollo también es conocido como *iPhone SDK*.

iOS SDK posee las interfaces, herramientas y recursos para desarrollar aplicaciones iOS. Este paquete sólo está disponible para sistemas Macintosh.

Apple ofrece la mayoría de su sistema de interfaces en paquetes especiales denominados frameworks. Los frameworks son directorios dinámicos que contienen librerías compartidas y los recursos necesarios para dar soporte a esas librerías. Para usar los frameworks en una aplicación, solo es necesario vincularla al proyecto. Al enlazarlas al mismo, se puede acceder a las características del framework y también a las herramientas de desarrollo.

Además de los frameworks, Apple también ofrece tecnologías librerías compartidas estándares. Ya que iOS está basado en UNIX, muchas tecnologías de bajo nivel derivan de tecnologías OpenSource, por lo tanto, éstas se encuentran disponibles en sus respectivos directorios.

Algunos de los componentes del SDK incluyen:

- *XCode*: es un entorno de desarrollo que administra los proyectos para cada aplicación en proceso. Permite editar, compilar, ejecutar y depurar el código. XCode es la principal aplicación utilizada durante el desarrollo.
- *iOS Simulator*: es una aplicación que simula la tecnología iOS, permitiendo probar los desarrollos localmente en una computadora Mac, sin necesidad de hacerlo en el dispositivo físico (iPhone/iPod/iPad).
- *Biblioteca del iOS Developer*: es una librería que contiene las referencias y documentación para el proceso de desarrollo.

---

<sup>18</sup>*Objective-C* es un lenguaje de programación orientado a objetos basado en el lenguaje C.

Aunque se pueden ejecutar y probar las aplicaciones en el iOS Simulator, XCode también permite correr y depurar los desarrollos directamente en los dispositivos conectados a la computadora, ya sea un iPhone, un iPod o una iPad. Esto permite probar rápidamente en un dispositivo real. Este tipo de prueba es más eficiente que el simulador, ya que hay ciertas características que el simulador no puede probar y en el dispositivo sí, como por ejemplo, el hardware de captura de videos e imágenes (cámaras integradas).

## **2.5.11. IOS Developer**

### **2.5.11.1.Introducción**

Quien quiere desarrollar para iOS puede hacerlo libremente, descargándose el SDK para desarrollo, junto con la guía de ayuda al usuario. Apple provee todas las herramientas necesarias para poder desarrollar sobre sus plataformas, el único requerimiento que es trabajar sobre sus productos. Por lo tanto, el desarrollo se debe hacer sobre cualquier producto de la línea de Apple (iMac, MacBook, MacBook Pro, MacBook Air, MacMini, etc.).

Apple provee un sitio denominado *iTunes Connect*, que permite administrar las aplicaciones que el desarrollador sube a la AppStore. Al desarrollar un proyecto de aplicación, al momento de subirlo, se debe generar un archivo denominado *binario*, que es la compilación de la aplicación, para que un equipo de revisión de aplicaciones lo testeé y determine que el producto se encuentra en correctas condiciones para ser publicado en la tienda.

### **2.5.11.2. Proceso de alta en AppStore**

Para subir una aplicación a la tienda de aplicaciones de Apple, conocida como AppStore, es necesario seguir una serie de pasos para que el proceso de publicación sea exitoso.

### **2.5.11.3. Asegurarse que la aplicación está lista para el proceso de aprobación**

#### ***2.5.11.3.1. Leer los lineamientos de revisión de AppStore***

Antes de enviar una aplicación para su aprobación, es necesario asegurarse que se cumple con el diseño técnico que se detalla en los *Lineamientos de Revisión de AppStore* y las *Directrices de Interfaz Humana*. Las aplicaciones que no siguen estas pautas serán devueltas para que sean modificadas nuevamente, antes de que puedan ser publicadas en la AppStore.

#### ***2.5.11.3.2. Probar en los dispositivos***

Es necesario probar la aplicación en su totalidad en los dispositivos en los que funcionará, sean iPad, iPhone y/o iPod touch. El simulador de dispositivos es ideal para mostrar los prototipos de la idea planteada en la aplicación, y poder visualizar cómo se va a presentar en el dispositivo.

Sin embargo, el simulador emula el sistema iOS, no el hardware, por lo que no es un sustituto para pruebas como lo son los dispositivos reales.

#### **2.5.11.4. Lo que se puede esperar durante la revisión de la aplicación**

El equipo de revisión de aplicaciones comprueba cada binario subido por los desarrolladores con el fin de proteger la privacidad de los consumidores, proteger a los niños frente a contenidos inapropiados y evitar las aplicaciones que degraden la experiencia central de los dispositivos iPad, iPhone e iPod touch. Todos los aspectos de las aplicaciones deben cumplir con los criterios establecidos en los Lineamientos de Revisión y debe cumplir con las Directrices de Interfaz Humana.

Después de que una aplicación ha sido revisada y aprobada, se establecerá en el estado *Ready for Sale*, lista para la venta, lo que indica que ya está disponible para ser accedida desde la AppStore.

Si un problema se descubre durante la revisión de la aplicación, se le notificará por correo electrónico a su desarrollador. Existe un apartado denominado *Centro de Resoluciones*, donde se detallan los motivos de los rechazos junto con otra información adicional proporcionada por el equipo de revisión de aplicaciones. Esto ayuda al desarrollador lo que debe tener en cuenta para corregir en la aplicación antes de ser subida nuevamente.

El periodo de revisión para aplicaciones es de generalmente 5 a 10 días hábiles. La mayor parte del tiempo, la aplicación se encuentra en estado *Waiting for Review*, en la cola de revisión por parte del equipo de Apple.

### 2.5.11.5. Comprobar el estado de la aplicación en iTunes Connect

Una vez que ha sido revisada y preparada para subir la aplicación (por el desarrollador), se debe crear la entrada en la página Administrar aplicaciones, en iTunes Connect. Esta entrada consiste en crear el espacio con los datos correspondientes que describirán y representarán la aplicación en la AppStore, el nombre de la aplicación, versión, la fecha de presentación, el Apple ID, así como el estado de la solicitud. Hay 16 estados indicados en tres colores (amarillo, verde y rojo) que pueden aparecer en la página administradora de aplicaciones, después de haber sido subida.



#### **Prepare for Upload**

Aparece como el primer estado de la aplicación. Este estado significa que se debe introducir o editar los metadatos, imágenes, precios, etc., para preparar la aplicación y subirla a la AppStore.



#### **Waiting for Upload**

Aparece cuando se ha terminado de introducir los metadatos y se indicó que está listo para subir el binario, sin embargo, no se ha terminado de cargar el binario al servidor. El administrador de aplicaciones está esperando que el binario sea subido para su revisión.



#### **Waiting for Review**

Aparece después de cargar la nueva aplicación o una actualización y antes de que la aplicación pase a revisión por parte de Apple. Este estado significa que la aplicación ha sido añadida a la cola de revisión de aplicaciones, pero aún no ha comenzado el proceso de revisión.



#### **In Review**

Aparece cuando Apple está revisando actualmente la aplicación, antes de cambiar al estado de rechazada o aprobada. Se necesita tiempo para revisar el binario, dependiendo de su complejidad, pero normalmente puede llevar 1 o 2 días.



#### **Pending Contract**

Aparece cuando una solicitud ha sido revisada y está lista para la venta, pero los contratos del desarrollador no están vigentes. Los contratos con Apple como desarrollador en iTunes Connect son anuales y deben ser renovados cada año para poder continuar formando parte de los programadores habilitados para AppStore.

 **Waiting For Export Compliance**

Si la aplicación tiene contenido con encriptación, se tendrá que esperar hasta que los CCATS (Commodity Classification Automated Tracking System) ha sido aprobado. Las leyes de exportación exigen que los productos que contienen cifrado estar debidamente autorizados para la exportación.

 **Upload Received**

Aparece cuando el binario se ha recibido a través de cargador de aplicaciones, pero aún no ha concluido el proceso dentro del sistema en iTunes Connect.. Este estado generalmente no dura más de algunos minutos.

 **Pending Developer Release**

Aparece cuando la versión de la aplicación ha sido aprobado por Apple y el desarrollador ha optado por establecer su control de lanzamiento de versiones. Se debe liberar la aplicación manualmente a la AppStore cuando esté lista.

 **Processing for AppStore**

Aparece cuando el binario está siendo procesado y estará listo para la venta dentro de las 24 horas en la AppStore.

 **Pending Apple Release**

Aparece cuando la versión de la aplicación se deba publicar por Apple en conjunto con la correspondiente versión del sistema operativo (iOS) liberado al público.

 **Ready for Sale**

Aparece una vez que la solicitud ha sido aprobada y publicada en la AppStore. Cuando la aplicación se encuentra en este estado, se tiene la opción de eliminarla de la tienda cuando el desarrollador lo crea conveniente.

-  **Rejected**

Aparece cuando el binario no ha pasado el examen. Se recibe una comunicación del equipo de revisión en el Centro de resoluciones, explicando el motivo del rechazo.
-  **Metadata Rejected**

Aparece cuando específica información en los metadatos, excluyendo el binario, no han pasado la revisión. Para resolver el problema, se tienen que editar los metadatos en iTunes Connect y el binario existente se volverá a utilizar para el proceso de revisión. Se recibe una comunicación del equipo de revisión en el Centro de resoluciones, explicando el motivo del rechazo de los metadatos.
-  **Removed from Sale**

Aparece cuando la aplicación ha sido retirada de la AppStore.
-  **Developer Rejected**

Aparece cuando se ha rechazado el binario del proceso de revisión por parte del desarrollador. Las versión existente de la aplicación no se verá afectada por el autorrechazo del binario. Esta acción pierde el lugar del binario en la cola de revisión, colocándola al final cuando se vuelva a enviar.
-  **Developer Removed from Sale**

Aparece cuando se ha eliminado la aplicación desde la AppStore. Este estado es cambiado por el desarrollador cuando desea quitar su aplicación de la AppStore.
-  **Invalid Binary**

Aparece cuando el binario se ha recibido en el sistema para revisión, pero no cumplió con todos los requisitos de carga. Se envía un correo electrónico detallando el problema y para poder arreglarlo antes de volver a subir el binario nuevamente.
-  **Missing Screenshot**

Aparece cuando la aplicación no contiene una captura de pantalla requerida para iPhone, iPod touch o iPad. Por lo menos una captura de pantalla se requiere para presentar la aplicación.

#### ***2.5.11.5.1. Notificaciones por correo de los cambios de estado***

En caso de configurarlo en iTunes Connect, se pueden recibir las actualizaciones de estado de la aplicación en la casilla de correo electrónico. De esta manera el desarrollador estará informado del estado de su aplicación en el momento que el equipo de revisión decida el mismo.

#### ***2.5.11.5.2. Historial de estado de la aplicación***

Se puede acceder a un registro con el historial de estado de la aplicación en iTunes Connect. Cada vez que se realiza un cambio de estado, se almacena el detalle. Se accede desde el administrador de aplicaciones mediante un enlace “Historial de estado”, donde se puede visualizar cada estado por el que la aplicación ha pasado, incluyendo la fecha en que se produjo el cambio. También se puede ver que usuario hizo el cambio en la cuenta iTunes Connect o si el cambio fue gestionado por Apple.

#### **2.5.11.6. Determinar la fecha de disponibilidad de la aplicación**

Puede establecer la fecha en que la aplicación estará disponible en la tienda de aplicaciones al momento de enviar el binario a revisión. Si la solicitud no ha sido aprobada por Apple antes de esta fecha, se publicará tan pronto como haya sido aprobada. Esta es una fecha global, y se aplica a todos los territorios seleccionados.

Con el control de liberación de versiones se puede controlar cuándo una actualización de la aplicación va a publicarse en AppStore, en lugar de publicarse tan pronto como sea aprobada por el equipo de revisión. Esta opción se administra al momento de subir el binario para ser revisado. Si se elige utilizar el control de liberación de versiones, el estado de aplicación va a cambiar a *Pending Developer Release* una vez que sea aprobada, lo que indica que se puede liberar a la AppStore cuando esté lista.

#### **2.5.11.7. Lineamientos de revisión para publicar en AppStore**

Para quienes desean invertir su tiempo y conocimiento para desarrollar aplicaciones para iOS, Apple ha publicado una guía de lineamientos que tiene en cuenta el equipo de revisión de AppStore para considerar apropiada una aplicación dentro de su tienda virtual. Esta guía permite clarificar las dudas y problemas de los desarrolladores mientras trabajan en sus aplicaciones (apps), y así acelerar el proceso de aprobación una vez que la misma esté terminada.

Apple ha decidido no permitir ciertos tipos de contenido en la AppStore, ya que tienen bien en cuenta que hay muchos niños descargando aplicaciones y los controles parentales no funcionan a menos que los padres lo configuren, y muchos no lo hacen.

Hay muchas aplicaciones para descargar en la tienda de Apple, alrededor de 700.000, por lo que existen diversos temas y utilidades ya realizadas. Apple exige que si la aplicación a realizar no es útil o no provee algún tipo de entretenimiento duradero, posiblemente no sea aceptada.

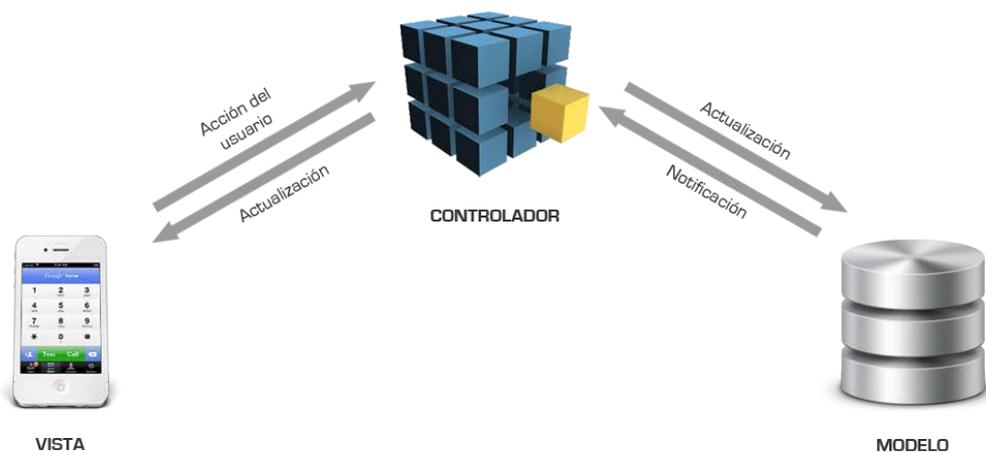
Apple es muy exigente en la calidad de los productos que ofrece, y bastante cauteloso en sus revisiones. En su guía publican: *“Rechazaremos las aplicaciones que muestren contenido o comportamiento que traspase la línea. Se preguntarán, ¿cuál línea? Pues, como la Suprema Corte de Justicia dijo una vez, ‘Lo sabré cuando lo vea.’ Y creemos que usted también se enterará recién cuando la cruce. (...) Si parecemos ser maniáticos del control, bueno, probablemente sea porque estamos muy comprometidos con nuestros usuarios, y nos aseguramos de que tengan una experiencia de calidad con nuestros productos. Tal como ustedes también lo hacen.”* (Apple Review Guidelines, 2013).

## 2.6. MVC

### 2.6.1. Definición y Estructura

El patrón MVC es un modelo de diseño en capas que se sigue para realizar un sistema de manera ordenada, siguiendo las 3 capas conceptuales: el *Modelo*, la *Vista*, y el *Controlador*. Es utilizado para modelar sistemas del tipo *cliente/servidor*. La siguiente imagen muestra el esquema:

*Imagen N°3: Representación del Modelo-Vista-Controlador.*



*Fuente: Elaboración propia. Fecha: 2012.*

El *Modelo* representa la información que utiliza la aplicación. Indica el modelo de la arquitectura de los datos y su comportamiento. El modelo responde a las peticiones solicitadas por la vista y actúa mediante los cambios generados en el controlador.

La *Vista* muestra la información de la aplicación al usuario. Esto lo realiza a través de interfaces gráficas, formularios, etc. Se muestra los datos procesados por el controlador de acuerdo a las necesidades particulares.

El *Controlador* responde a los eventos que se solicitan desde la vista. Utiliza la lógica de negocio para procesar y entregar una respuesta. Es el nexo entre la vista y el modelo.

### 2.6.2. Uso en iOS

Para realizar aplicaciones en iOS, Apple sugiere de manera predeterminada el uso del patrón MVC. No solo se definen los objetos que interactuarán, sino también la manera en la que lo harán. De esta manera se puede realizar un buen diseño de aplicación, consistente y de correcta comunicación.

Particularmente, las tareas de cada objeto, son:

- *Objetos de Modelo*: se encarga de encapsular los datos contenidos en la aplicación. Por ejemplo, la aplicación de libreta de contactos contiene contactos almacenados que pertenecen al modelo, que en este caso, son datos persistentes.
- *Objetos de Vista*: es la vista que el usuario puede ver. Contiene los patrones de diseño de la misma, es decir, se encarga de su propio dibujo y es encargado de recibir las acciones del usuario, mostrar y permitir ingreso de datos
- *Objetos de Controlador*: las acciones que el usuario realiza en la capa de vista, crean o modifican datos que se comunican a través de un objeto controlador, cuyo resultado es una consulta o modificación del modelo. Este es el intermediario entre la vista y el modelo.

Se pueden combinar los roles desempeñados en objetos MVC, por lo que un objeto puede cumplir con roles de controlador y vista al mismo tiempo. En este caso se trataría de un objeto *ViewController* (vista-controlador). Para algunas aplicaciones, la combinación de este tipo es un diseño aceptado.

## 2.7. WebServices

### 2.7.1. Definición

Debido a la gran diversidad de tecnologías que se utilizan en el entorno web particularmente, se genera el pequeño problema de la comunicación por internet entre aplicaciones de diferentes lenguajes en computadoras de hasta diferentes sistemas operativos. Actualmente, y para seguir el hilo, se deben tener en cuenta que las comunicaciones también pueden ser realizadas desde dispositivos móviles como smartphones y tablets.

Para definir, los webservices son sistemas que permiten la comunicación entre diferentes aplicaciones para compartir información entre distintos dispositivos y servidores ubicados geográficamente distantes, a través de la web, utilizando estándares como XML, SOAP, WSDL, UDDI sobre protocolos de internet. Los webservices permiten la posibilidad de conectar sistemas ejecutados en diferentes plataformas, corriendo en diferentes sistemas operativos, escritos en diferentes lenguajes de programación y ubicados en diferentes lugares, de una manera estandarizada y sencilla de aplicar. Estos estándares cumplen determinadas tareas:

- *XML (eXtensible Markup Language)*: es un lenguaje de anotación extendido similar al lenguaje HTML que fue diseñado para estructurar, almacenar e interpretar datos.
- *SOAP (Simple Object Access Protocol)*: es un protocolo para el intercambio de información entre aplicaciones a través de internet, utilizando HTTP. Este protocolo es independiente de la plataformas que se estén usando y del lenguaje en que se hayan escrito las aplicaciones que interactúan.
- *WSDL (Web Services Description Language)*: es un documento XML utilizado para describir y localizar los servicios web utilizados, así como las operaciones o métodos que el mismo maneja.
- *UDDI (Universal Description Discovery and Integration)*: es el modelo o distribución estandarizada de directorios que utilizan los webservices. UDDI utiliza WSDL para describir las interfaces de los mismos. En pocas palabras, es el directorio para almacenar información de los webservices.

### **2.7.2. Utilidad**

Los webservices son muy importantes hoy en día, si bien para conectar diferentes sistemas, también para utilizar funciones de unos con otros. De esta manera se aprovecha y reutiliza código, y se generan y mantienen el o los sistemas con sencillez.

Un ejemplo claro del uso efectivo de webservices es en sitios de e-commerce. Generalmente una aplicación que realiza transacciones bancarias, no lo hace directamente, sino que el banco brinda, a través de un webservice, una interacción con el sistema propio para poder realizar pagos, transferencias, verificación de tarjetas de crédito, etc. El banco actúa de intermediario entre la transacción y el sistema creado (a través de su propio sistema bancario) y facilita que nuestro sistema tome esa información a través de un webservice. Para el usuario final, todo este proceso es transparente.

### **2.7.3. Relación con App iOS**

La aplicación en marcha para los servicios académicos de la Universidad del Aconcagua, utiliza webservices que permiten a la aplicación conocer todos los cambios que se van realizando periódicamente, desde el sector de administración. El sistema web permite gestionar los datos necesarios de manera simple, y a través de la aplicación nativa móvil se pueden ver reflejados dichos cambios.

Por otra parte, la aplicación debe interactuar con una base de datos web, como es MySQL, lo cual, por limitaciones, no puede hacerlo directamente. Es por eso que el uso de webservices son fundamentales para permitir esta conexión o enlace de manera casi directa.

## **2.8. Seguridad y Autenticación**

Cuando se realiza una conexión entre un cliente y un servidor donde se necesita trabajar con datos de información personal y/o confidencial, se hace necesario generar del lado del cliente una autenticación que permita al servidor constatar que el usuario es el propietario de esos datos; mientras que, del lado del servidor, se necesita certificar que la empresa, el sitio o el servidor al que el usuario se está conectando no es una entidad falsa y es realmente quien dice ser.

La autenticación sencillamente consiste en que el usuario verifique su identidad. Esto se hace mediante un control de acceso verificado por un usuario y una contraseña asociada al mismo, previamente creados por el usuario en cuestión. Estos paneles de control de acceso se los conoce como login. En algunos casos, donde se requiere mayor información para la validación del usuario, se solicitan más datos, como por ejemplo, número de documento (DNI), código de seguridad, frase específica, etc.

Para proteger la información de los usuarios, existen tecnologías llamadas "SSL" (Secure Sockets Layer, Capa de Conexión Segura) y "TLS" (Transport Layer Security, Seguridad de Capa de Transporte) que encriptan los datos que se transmiten entre un navegador web y un servidor web. Las direcciones web protegidas con SSL o TLS comienzan con `https://` en lugar de `http://`. Algunas personas se refieren directamente como "HTTPS".

## **2.9. Diseño de Interfaces**

### **2.9.1. Mockups**

Con la llegada de los dispositivos móviles, aparece un nuevo enfoque en el diseño de interfaces y presentación: los mockups.

Básicamente, los mockups son maquetas que tienen el objetivo de mostrar un boceto preliminar de diseño y usabilidad de la aplicación en desarrollo. También, mediante el storyboard, las secciones, opciones, secuencias disponibles que la aplicación posee. Este concepto muestra la navegación que se puede experimentar.

Los mockups permiten dividir las tareas de acuerdo a las pantallas que se diseñan, de acuerdo a los componentes que se quieren incluir, y también permiten detectar errores en el diseño, posibilitando incorporar cambios de manera sencilla.

Estos prototipos de interfaces también son aplicados en el entorno web, por lo que los diseños de sitios y páginas utilizan estos modelos, permitiendo identificar cada pantalla con sus componentes y describiendo sus detalles.

Existen aplicaciones que permiten realizar diseños de pantallas, tanto de aplicaciones web, como también de móviles, aunque una manera sencilla de representarlas es diseñando sobre un

boceto específico de dispositivo, detallando las medidas para ser preciso al momento de desarrollar ya sea componentes y espacios disponibles, como también imágenes que cada pantalla presenta.

Se presenta la plantilla para el diseño de las interfaces y pantallas en un dispositivo móvil iPhone (ver Anexo B).

## **2.10. Mobile Apps y las Universidades**

En otras universidades alrededor del mundo se han realizado aplicaciones que permiten al alumno interactuar con su universidad. La mayoría puede verse en EEUU, aunque en varios países las universidades se han animado a lanzar sus propias aplicaciones para permitir al alumno mantenerse informado y acceder al contenido específico facilitado por esta plataforma.

Algunos ejemplos de aplicaciones de universidades pueden ser las siguientes detalladas a continuación:

### **2.10.1. Universidad de Granada: Granada, España.**

La Universidad de Granada presenta una aplicación para dispositivos móviles de la familia iOS. Esta aplicación provee información útil sobre la Universidad, sus órganos de gobierno actuales, los centros que la componen y las titulaciones que en ella se imparten.

Con esta aplicación se puede conocer la Universidad y la oferta que ofrece a la comunidad educativa.

La aplicación dispone también de información útil para alumnos extranjeros o foráneos, como es el alojamiento en Granada, así como también el calendario académico, el uso de la tarjeta Universitaria o el proceso para solicitar una cuenta de correo electrónico.

La aplicación incluye acceso al directorio de la Universidad, con el que se puede localizar la información sobre los miembros de esta comunidad.

También se puede localizar geográficamente los centros y servicios de la Universidad.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°4: Aplicación móvil de la Universidad de Granada.*



*Fuente: AppStore, Apple. Fecha: 2012.*

## 2.10.2. Universidad de Navarra: Navarra, España.

La aplicación de la Universidad de Navarra para smartphones y tablets ofrece información acerca de la oferta académica de la Universidad, sus Grados, Másteres y Programas de Doctorado. Los futuros alumnos disponen de vídeos tutoriales que explican de forma gráfica los aspectos fundamentales a tener en cuenta en procesos como la búsqueda de alojamiento, becas y ayudas, o admisión. Ofrece información y mapas de los 5 Campus de la Universidad de Navarra y un

completo directorio con los datos de contacto de sus Facultades, Centros y Departamentos. También permite acceder a las noticias de la Universidad, los eventos, y todos los contenidos multimedia en vídeo y los podcast de su radio (98.3).

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°5: Aplicación móvil de la Universidad de Navarra.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.3. Universidad de Alicante: San Vicente del Raspeig, Alicante, España.

La Universidad de Alicante ha creado esta aplicación para ofrecer la mayor información posible. La aplicación, actualmente, es universal, pudiendo trabajar de forma nativa en el iPad, en el iPhone y en el iPod.

Desde la aplicación se puede:

- Acceder a las noticias de las facultades y de la propia universidad.
- Conocer los nuevos planes de estudio de grados y los nuevos masters.
- Acceder a la información del directorio de la universidad con la posibilidad de llamar directamente, escribir un email o de cargar la información de contactos.

- Acceder a los vídeos creados por la universidad.
- Saber la ubicación y donde se encuentran los distintos edificios de la universidad, así como cada uno de sus campus.
- Información institucional de la universidad.
- Acceso a varias de las funciones disponibles en el Campus Virtual de la universidad, como:
  - Anuncios
  - Materiales
  - Tutorías
  - Controles
  - Enlaces
  - Horario de clase
  - Fecha de los exámenes
  - Ficha de las asignaturas
  - Expediente académico

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°6: Aplicación móvil de la Universidad de Alicante.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.4. Universidad de Jaén: Andalucía, España.

La Universidad de Jaén también apuesta por las nuevas tecnologías móviles presentando su aplicación para la plataforma iOS.

A través de este soporte se puede acceder a todo tipo de contenido relacionado con la Universidad:

- Conocer las últimas noticias de nuestro diario digital.
- Consultar de la oferta académica.
- Guías a través del campus.
- Contenido multimedia con los que se puede conocer más a fondo todas las actividades que se producen en nuestro campus.
- Información de contacto de todos los servicios de la Universidad, tales como departamentos, centros, etc.

Además incluye la posibilidad de escuchar la emisora su radio en directo.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°7: Aplicación móvil de la Universidad de Jaén.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.5. Universidad de Phoenix: Estados Unidos.

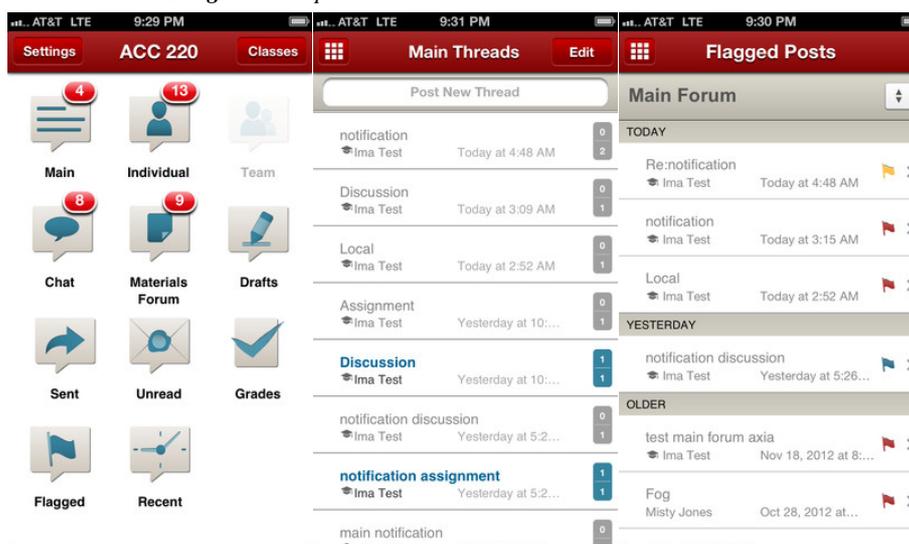
La aplicación de la Universidad de Phoenix ha mejorado con el tiempo. Ahora se puede acceder al chat y materiales, foros, ver archivos adjuntos y acceder a la misma sin conexión.

Phoenix Mobile tiene las características que necesitan los estudiantes:

- Ver información publicada por profesores en el foro de materiales.
- Abrir archivos adjuntos.
- Publicar en foros de discusión mensajes para compañeros, editar y guardar borradores.
- Navegar a través de foros, guardar y eliminar los borradores.
- Participar en los foros de discusión en línea en clase.
- Ver indicadores de mensajes, preguntas de discusión y anuncios de clase.
- Recibir alertas cuando se reciben mensajes y/o respuestas de temas y cuando se publican las calificaciones.
- Navegar foros, incluso si se está fuera de línea.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°8: Aplicación móvil de la Universidad de Phoenix.*



*Fuente: AppStore, Apple. Fecha: 2012.*

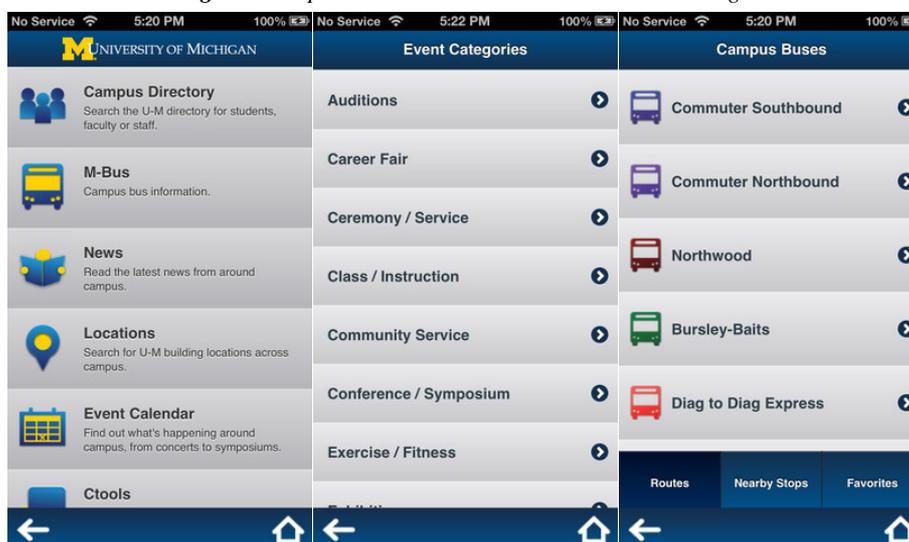
### **2.10.6. Universidad de Michigan: Ann Arbor, Michigan, Estados Unidos.**

La aplicación de la Universidad de Michigan contiene las siguientes características:

- Buscar en el directorio de mensajería unificada para estudiantes, profesores o personal. Permite agregar automáticamente nuevos contactos al teléfono.
- Ver en el mapa las rutas de autobús, buscar una parada y saber cuando llegará el próximo autobús. El mapa está animado en tiempo real para mostrar el progreso autobuses a cada parada.
- Leer las últimas noticias del diario de Michigan, la University Records, UM News Service, el Sistema de Salud de la UM, etc.
- Buscar lugares de construcción de UM en el campus.
- Calendario de Eventos. Averiguar lo que está sucediendo en el campus, desde conciertos a reuniones o conferencias.
- Revisar el curso y los sitios de los proyectos sobre la marcha. Ver notas, recursos de acceso, leer anuncios, etc.
- Servicios de restauración. Ver menús para todos los comedores del campus para el día y el resto de la semana.
- Localizar un asiento libre en los laboratorios de computación del campus.
- Buscar en el catálogo de la biblioteca y usar otros recursos de la biblioteca móvil.
- Visitar el sitio móvil de la Universidad.
- Contactos de emergencia. Ver la policía local y los números de teléfono de atención médica cercanos.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°9: Aplicación móvil de la Universidad de Michigan.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### **2.10.7. Universidad de Miami: Miami, Estados Unidos.**

UMiami Mobile es la aplicación de la Universidad de Miami, que permite tener acceso instantáneo a las noticias y eventos, información del curso, horario de los deportes y las puntuaciones, videos, etc.

- Directorio: permite el acceso total al directorio público, para llamar y enviar por correo electrónico a prácticamente cualquier persona en todo el campus, y añadirlos directamente a la libreta de direcciones del teléfono.
- Atletismo: se puede consultar horarios, resultados y mantenerse al día con los deportes de la Universidad.
- Eventos: permite encontrar y mantener un seguimiento de una amplia gama de actividades de la escuela y ex alumnos, desde conciertos y exposiciones de arte, de conferencias y de recreación.

- Cursos: se puede navegar fácilmente por el Departamento a través del catálogo de cursos de la Universidad.
- Mapas: permite encontrar fácilmente la ubicación exacta, la búsqueda de un edificio y determinar su dirección.
- Tránsito: realiza el seguimiento del autobús de la Universidad que se mueve sobre su recorrido en el campus.
- Videos: permite acceder a contenidos de video de la Universidad.
- Noticias: contiene noticias y artículos actualizados sobre lo que está sucediendo en torno a la Universidad.
- Imágenes: se puede examinar o buscar imágenes de la Universidad. También guardar imágenes en el iPhone o iPod Touch.
- Radio: contiene la estación de radio de la Universidad.
- Estacionamiento: permite ver la disponibilidad en tiempo real de los estacionamientos que contiene la Universidad.
- Emergencias: facilita los números de emergencia dentro o fuera del campus.
- Encontrar un doctor: modulo para encontrar a un médico cercano.
- Webcams: se puede visualizar el campus a través de las cámaras web colocadas en la Universidad.
- Librería: permite comprar todo lo que el alumno necesita a través del enlace Bookstore.
- Comedor: presenta lo que hay en el menú de todas las instalaciones del comedor en el campus.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°10: Aplicación móvil de la Universidad de Miami.*



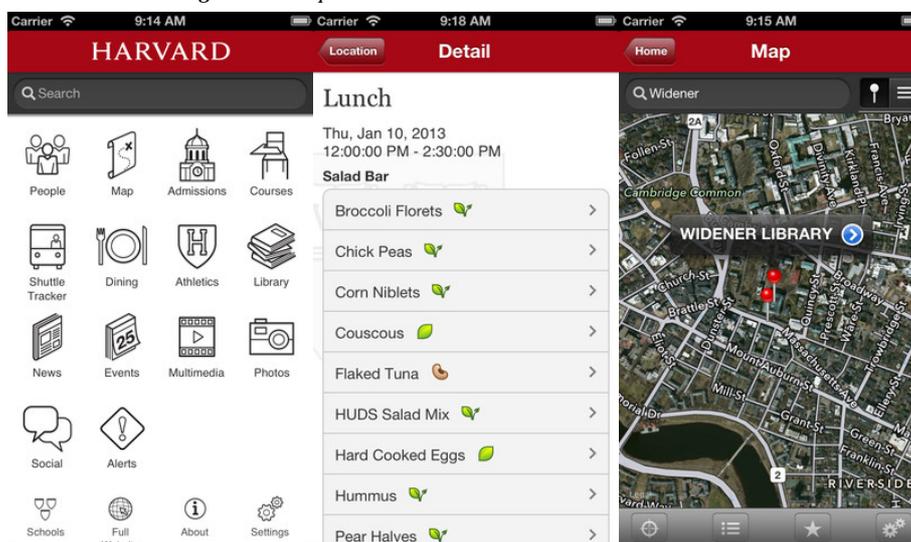
*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.8. Universidad de Harvard: Cambridge, Massachusetts, Estados Unidos.

La aplicación de la Universidad de Harvard es parte de una iniciativa más amplia para mejorar la experiencia móvil de los estudiantes, profesores, personal, visitantes y vecinos que se relacionan con la escuela y la comunidad de Harvard.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°11: Aplicación móvil de la Universidad de Harvard.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.9. Universidad de San Diego: San Diego, California, Estados Unidos.

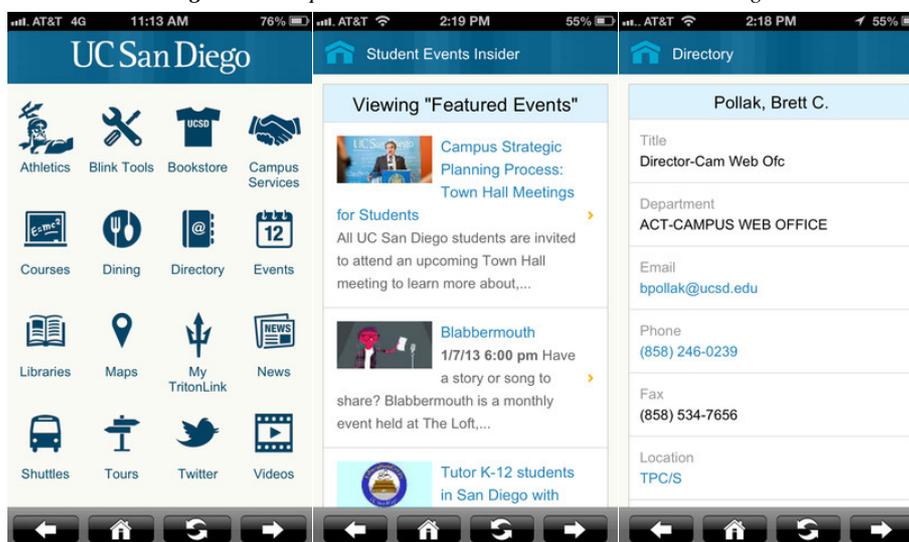
La aplicación de la Universidad de San Diego tiene servicios para encontrar información de contacto para los alumnos, encontrar el camino a través del campus, leer noticias del campus, revisar los resultados deportivos, ver videos relacionados con el campus, encontrar información de autobús, obtener una visión general de los cursos actuales y futuros cursos programados y escuchar audios de conferencias populares.

### Características:

- MyTritonLink: ver el horario de clases, con rutas de senderismo y mapas personalizados.
- Directorio: consultar los estudiantes, profesores y personal por nombre o apellido, para ponerse en contacto con la gente por teléfono y correo electrónico.
- Mapas: buscar en los edificios del campus por nombre, señalar que en el mapa, ver las fotos de construcción, y ver la ubicación deseada.
- Traslados: ver localización en tiempo real y llegadas estimadas de autobuses del campus, incluyendo rutas fuera del campus.
- Noticias: navegar a través de varias fuentes de noticias del campus, incluyendo la investigación, el Centro Médico y las bibliotecas.
- Cursos: examinar y buscar el horario del curso actual y futuro a largo plazo, incluyendo descripciones, horarios, lugares. Incluso se puede aprovechar para encontrar el lugar del curso en el mapa o en contacto con el profesor.
- Audio: bajo el icono de cursos, escuchar audios de los cursos en el dispositivo móvil.
- Videos: revisión y visualización de videos desde el canal de YouTube de la Universidad de San Diego.
- Atletismo: ver noticias deportivas de la Universidad, horarios, y las puntuaciones de partidos.
- Bibliotecas: ver horas, ubicaciones y buscar en el catálogo.
- Tours: ver recorrido virtual auto-guiado de algunos hitos destacados del campus o registrarse para un viaje en persona desde el dispositivo móvil.
- Comedor: ver comedores del campus, horas y menús diarios de las instalaciones gestionadas por cada comedor.
- Librería: buscar libros de texto por el término / sección, encontrar información actualizada sobre las ventas y eventos de la librería.
- Twitter: tweets de la Universidad recogidos desde un solo lugar.
- Clima: ver las condiciones climáticas actuales del campus y las predicciones de cinco días.
- Eventos: ver el calendario de eventos, eventos estudiantiles, las fechas académicas y administrativas.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°12: Aplicación móvil de la Universidad de San Diego.*



*Fuente: AppStore, Apple. Fecha: 2012.*

#### **2.10.10. Universidad de Bradford: Bradford, Reino Unido.**

La aplicación de la Universidad de Bradford añade funcionalidades todo el tiempo, lo que la hace una herramienta necesaria para los estudiantes, empleados y personas vinculadas a la Universidad.

Las características de la aplicación son:

- Búsqueda de campus o de edificios, servicios, y otros lugares a través de Google Maps;
- Encontrar las habitaciones que tengan PC libres más próximas a través de la ubicación de GPS;
- Seguimiento de amigos mediante su ubicación;
- Búsqueda de teléfonos y correos electrónico de contactos;
- Recepción de alertas y avisos importantes;
- Búsqueda en el catálogo de la biblioteca, reserva de libros, y verificación de información de cuenta de estudiantes;

- Comprobación de los accesorios deportivos;
- Guía de bolsillo para la Universidad;
- Radio estudiantil y canal YouTube de la Universidad;

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°13: Aplicación móvil de la Universidad de Bradford.*



*Fuente: AppStore, Apple. Fecha: 2012.*

### 2.10.11. Universidad de Liverpool: Liverpool, Reino Unido.

La aplicación permite el acceso móvil a:

- Intranet: una sesión para ver correos electrónicos, calendarios, horarios y otra información personalizada.
- Asientos de PC: encontrar PC libres más cercanas.
- Novedades: noticias de gremios asociados.
- Aprender: obtener acceso instantáneo a todo el contenido vital en el camino. Revisión de calificaciones, publicar en foros de discusión, ver los avisos de clase importantes, leer y descargar las tareas, subir contenido, y mucho más.

- Mapa Campus: averiguar la ubicación actual y donde tiene que ir.
- Directorio: buscar el directorio para los estudiantes, el personal docente y los administradores de datos de contacto. Enviar un correo electrónico, hacer una llamada y guardar información de contacto a la libreta de direcciones del teléfono.
- Eventos: ver lo que está pasando cerca de la ubicación actual, encontrar lo que está ocurriendo en la Universidad o alrededores, ver conciertos, conferencias, exposiciones, etc.
- Noticias: obtener las últimas noticias, navegar y buscar artículos recientes para estar al tanto de las noticias más importantes de la Universidad.
- Imágenes: buscar, navegar, descargar y compartir imágenes de alta calidad de la Universidad.
- Videos: acceso rápido y fácil al canal de YouTube de la Universidad, eventos y audios disponibles para ver en cualquier momento.
- Ayuda: acceder a los números importantes cuando se necesitan. Llamar para reportar una emergencia.
- Estado del servicio: comprobar el último estado de los sistemas de la Universidad, correos electrónicos, etc.
- Cursos: Buscar en la lista de cursos o utilizar la herramienta de búsqueda para encontrar el curso solicitado por el estudiante.
- Biblioteca: Examinar el catálogo de la biblioteca completa, consultar la disponibilidad de artículos, revisar la cuenta del estudiante y ver el horario.

A continuación se muestran algunos screenshots de la aplicación:

*Imagen N°14: Aplicación móvil de la Universidad de Liverpool.*



*Fuente: AppStore, Apple. Fecha: 2012.*

## 2.11. Síntesis

Como se ha descrito anteriormente, las instituciones académicas, así como también las grandes (y también algunas pequeñas y medianas) empresas, necesitan amoldarse a los cambios de plataformas, entre otras cosas, para mejorar y facilitar sus prestaciones a sus usuarios, así como también para expandir su mercado y evitar perder sus actuales clientes, quienes a su vez están migrando a la movilidad. La inclusión de la Universidad dentro del mercado móvil es una buena estrategia para incrementar la popularidad de la misma.

Si bien no se puede hablar de un reemplazo y técnicamente se está lejos de ello, los smartphones ocupan una porción importante sobre el uso cotidiano que el usuario común requiere. Se dice que los smartphones y su evolución han transformado el comportamiento de los consumidores, de modo que las tareas que antes realizaban ya no son enfrentadas de la misma manera. Los usuarios realizan tareas diarias sin necesidad de localizarse en un lugar fijo, sino en gran parte en movimiento o transportes públicos entre otros, situación que no es habitual con otros dispositivos de mayores dimensiones, como por ejemplo, una notebook.

### **3. Desarrollo**

#### **3.1. Tecnologías**

##### **3.1.1. Versión de XCode**

En el presente desarrollo se utiliza XCode versión 4.6, que agrega la implementación del desarrollo orientado a dispositivos con pantalla retina.

Para el desarrollo sobre esta tecnología se necesita mínimamente una computadora Macintosh con Lion, ya sea una MacMini, iMac, MacBook, MacBook Pro o MacBook Air, dado que el Xcode 4.6 se ejecuta en los sistemas operativos Mac OS X 10.8 (Mountain Lion) y Mac OS X 10.7 (Lion).

Desarrollar el proyecto en esta versión de XCode permite trabajar en un sistema que pueda funcionar en dispositivos con sistema operativo iOS desde la versión 4.3 hasta la 6.1, abarcando así la mayoría de los dispositivos del mercado. La mayoría de los usuarios actualmente utiliza iOS5.

### 3.1.2. Lenguaje PHP

El lenguaje de programación que se utiliza para el desarrollo del sistema es PHP en su versión 5. Tanto el sistema de edición y administración de la Universidad, así como también los servicios web que utiliza la aplicación móvil están realizados en código PHP orientado a objetos.

Se utiliza PHP por ser un lenguaje de scripting especialmente adecuado para el desarrollo web, que puede ser embebido en páginas HTML para obtener contenido dinámico. PHP permite el acceso a información almacenada en diferentes bases de datos, como por ejemplo MySQL, SQLite, MSSQL, etc., lo que lo hace un lenguaje potente y respetable para desarrollos de alta magnitud.

### 3.1.3. Servidor Web

La opción a utilizar como servidor web es Apache, que es un servidor web que implementa el protocolo HTTP. Se utilizó esta opción por poder ser utilizado en diferentes sistemas operativos, tales como Windows, GNU/Linux y Mac OS.

El servidor Apache es del tipo *Open Source*<sup>19</sup> y puede ser utilizado de manera libre.

Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. El front end del motor de búsqueda Google está basado en una versión modificada de Apache. Muchos proyectos de Wikimedia también se ejecutan sobre servidores web Apache.

---

<sup>19</sup>*Open Source (Código Abierto)*: La idea del código abierto se centra en la idea de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario.

### 3.1.4. Bases de datos

Dentro de la aplicación nativa del dispositivo móvil, para que los datos persistan, deben almacenarse en una base de datos local. De esta manera, el usuario no necesita mantener una conexión a internet cada vez que requiera utilizar y/o consultar la aplicación.

#### 3.1.4.1. *SQLite*

La utilización de SQLite es fundamental para el uso en iOS, dado que tiene alta rapidez y por tanto, lo hace muy útil para el uso en dispositivos portátiles (móviles).

La aplicación utiliza un framework adicional para manejar las bases de datos del tipo SQLite, contenido en los recursos de la herramienta XCode, que se debe importar en cada proyecto que se vaya a utilizar bases de datos. La librería se encuentra con el nombre de *libsqlite3.0.dylib*. Se agrega al proyecto desde el editor XCode, en la sección de frameworks y pesa alrededor de 1,7 MB.

#### 3.1.4.2. *MySQL*

MySQL es elegido por su motor de base de datos relacional. MySQL es muy utilizado en aplicaciones web, ya que es muy rápida en cuanto a lectura de datos. En aplicaciones web hay generalmente baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en

lectura de datos, lo que hace a MySQL ideal para esta aplicación. Por consiguiente, el servidor que aloja el sitio administrador contiene una base de datos del tipo MySQL.

### 3.1.5. Webservices

Específicamente, para la comunicación de los webservices se utiliza el protocolo SOAP que permite realizar la comunicación intercambiando datos por medio de XML.

Los webservices permiten que se establezca la comunicación entre una aplicación web y la aplicación móvil, permitiendo al dispositivo llegar a los datos del servidor y traer los necesarios, o los solicitados.

Existe un conjunto de clases PHP que permite crear y consumir servicios web basados en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. Esta librería se llama *NuSOAP*. Para el desarrollo es imprescindible el uso de NuSOAP, con el que se realiza la comunicación entre el servidor de datos del sistema y la aplicación móvil, de esta manera se origina un enlace entre los datos del servidor con el dispositivo, permitiéndole al mismo realizar consultas en la base de datos remota y procesar y guardar lo necesario en su base de datos local.

NuSOAP realiza la comunicación y envío de mensajes por medio del lenguaje XML, y tiene un formato específico para que sea interpretado:

```
<?xml version="1.0"?>

<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
```

```

<soap:Fault>
...
</soap:Fault>

</soap:Body>

</soap:Envelope>

```

Las partes específicas del mensaje anterior se explican a continuación:

```
<?xml version="1.0"?>
```

Al ser un documento XML, debe comenzar con el tag `<?xml...?>` junto con la versión correspondiente.

```
<soap:Envelope
```

Se especifica que comienza el contenido del mensaje (envelope o sobre).

```
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

El mensaje SOAP contiene un elemento `envelope` asociado con el namespace: `http://www.w3.org/2001/12/soap-envelope`

```
Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

Indica donde se encuentran definidos los tipos de datos utilizados en el documento.

```
<soap:Header>
```

Se especifica el comienzo del Header (encabezado). Aquí se incluye información específica del mensaje.

```
</soap:Header>
```

Indica la finalización del Header (encabezado).

```
<soap:Body>
```

Comienza el cuerpo del mensaje, donde se incorpora toda la información necesaria para el nodo final. Por ejemplo, los parámetros para la ejecución, o la respuesta a una petición.

```
<soap:Fault>
```

Cualquier tipo de fallo producido se notifica dentro del tag `Fault`, cual está contenido dentro del cuerpo del mensaje (`Body`).

**</soap:Fault>**

Cierre del tag Fault.

**</soap:Body>**

Indica el final del cuerpo del mensaje.

**</soap:Envelope>**

Finaliza el mensaje SOAP.

## **3.2. Seguridad y Autenticación**

Para mantener el sistema protegido y confiable ante los ingresos externos, se plantea instalar un certificado SSL, el cual permitirá verificar que el sitio forma parte de la Universidad del Aconcagua.

### **3.2.1. Configurar SSL en el sitio web**

Para implementar en una o más páginas de el sitio web, es necesario realizar los siguientes pasos:

Obtener un certificado SSL para el sitio web, que es el documento electrónico que verifica la identidad de una empresa y permite que un servidor web establezca una encriptación segura con el navegador web del usuario visitante. Al igual que el alojamiento de sitios web, hay muchas entidades proveedoras de certificados SSL disponibles, como por ejemplo, VeriSign, CertiSur, etc.

Instalar el certificado SSL en el servidor web. El método de instalación varía según el servidor web y el tipo de certificado adquirido. Normalmente, el proveedor del certificado o ofrece ayuda, recursos e instrucciones específicas de instalación.

Identificar las páginas del sitio web que se desea asegurar con SSL. Los sitios web más seguros usan SSL en todo el sitio. Sin embargo, se puede aplicar sólo en las páginas que usan conexiones seguras o en páginas que recopilen o transmitan información personal y financiera determinada, como contraseñas personales de acceso, información de contacto o números de cuentas bancarias.

Modificar los vínculos o enlaces a las páginas (y los elementos de las páginas) que se quieren cargar de forma segura. En el caso de las páginas que se cargarán de forma segura, se deben cambiar los vínculos a esas páginas para que al principio incluyan `https://` en vez de `http://`. También es recomendable que se configure los redireccionamientos del servidor para que dirijan automáticamente a los usuarios que intentan visitar una URL insegura.

Realizar una prueba para verificar que las páginas sean seguras. Visitar todas las nuevas páginas seguras mediante dos navegadores diferentes como mínimo, que los usuarios visitantes normalmente podrían usar. Si se ve un ícono de candado en el navegador, es señal que la página está configurada correctamente. Se puede verificar haciendo clic en él para obtener más información que confirme que las conexiones son seguras. El error más común es tener "contenido combinado" en una página `https://`. Esto quiere decir que uno o más elementos (generalmente imágenes, archivos flash o archivos CSS) se cargan en una página `https:` con una URL `http://` que no es segura. Los navegadores más modernos indican los recursos inseguros en páginas de contenido combinado en la consola de JavaScript (en algunos navegadores, se puede llamar "depurador de JavaScript"). Para solucionar estos problemas hay que examinar el código HTML de la página y realizar lo siguiente:

- Buscar en el código: `"http://"`.
- Reemplazar todas las instancias que encuentre por `"https://"`.
- Guardar los cambios en el servidor web y volver a realizar pruebas.
- Si se siguen recibiendo advertencias de contenido combinado en la página, es muy probable que se deba a problemas con el código de Javascript o flash.

### **3.2.2. Costo de SSL**

Los precios de los servicios de SSL varían dependiendo la entidad certificante. A marzo de 2012, muchos proveedores de certificados SSL ofrecían los mismos menos de 100 dólares por año. Muchos webhostings de comercio electrónico ofrecen servicios básicos de SSL como parte de paquetes de alojamiento web sin ningún costo adicional.

### **3.3. iOS Developer**

Se creará una cuenta iOS Developer dentro del portal de desarrolladores de Apple, que dará acceso a la cuenta para la facultad del Aconcagua, la cual le permitirá gestionar la aplicación móvil, así como también contenidos, estadísticas, cobros (si se llegase a requerir) y otras aplicaciones que puedan crearse en el futuro.

Como desarrollador de aplicaciones para iOS, se deben respetar ciertas condiciones que serán consideradas a la hora de revisar cada proyecto. El documento “*Apple Review Guidelines*” indica las normativas que se deben tener en cuenta para desarrollar una app iOS (ver Anexo A).

## **4. Aplicación**

El desarrollo del sistema requiere una serie de pasos para la producción del producto con la finalidad de tratar a la problemática planteada anteriormente, ayudando a mejorar los procedimientos administrativos presentados, mediante las herramientas informáticas.

### **4.1. Investigación**

El proceso de investigación se lleva a cabo mediante la técnica de observación directa, a fin de recopilar información sobre el sistema web utilizado por la Universidad, tratando de capturar el mayor número de datos necesarios.

Esta técnica se implementó realizando un seguimiento y haciendo un uso minucioso del sitio de la Universidad, utilizando particularmente las herramientas referentes a los servicios online proporcionados al alumno.

También se observó cuidadosamente el movimiento en las oficinas físicas de secretaría, tanto en tareas administrativas, como en flujo de alumnos.

Otra revisión que se realizó, fortaleciendo la investigación y para tener en cuenta al momento de diseñar el sistema, es la visita a diferentes sitios universitarios del mundo, cuyos sistemas ya tuvieran implementadas aplicaciones orientadas a móviles. Este estudio permitió analizar el impacto de las mismas, las características que pueden ser útiles al presente desarrollo y las reseñas de los usuarios, en este caso, alumnos de sus respectivas instituciones. Esta revisión es sólo superficial, ya que no se puede tener acceso a los sistemas de universidades de las cuales no se es miembro, pero se pueden ver, por ejemplo, en la tienda de Apple, las diferentes características e implementaciones que se han realizado para esas aplicaciones y, de acuerdo a los usuarios, si éstas son favorables o no.

## 4.2. Requerimientos

Los requerimientos o características que deben incluirse en el nuevo sistema están orientados a captar o procesar datos, producir información, controlar una actividad o dar apoyo a una tarea.

De acuerdo a la investigación previamente realizada y basándose en la idea planteada del sistema, el mismo se divide en 2 partes esenciales: la sección administrativa y la de consulta.

La parte administrativa refiere al mantenimiento, que es manejado por el sector administrativo de la institución, más precisamente los empleados del Área de Secretaría de la Universidad.

Con respecto a la parte de consulta, probablemente la más utilizada, es dedicada particularmente a los usuarios comunes de la Universidad, es decir, los estudiantes. Si bien los alumnos pueden llegar a tener otras acciones diferentes a las de consultas, se considera que la mayoría son del tipo de “sólo lectura”.

El personal administrativo, entre otras tareas, es el responsable de cargar la siguiente información:

- Facultades de la Universidad, esta tarea se realiza una sola vez o muy pocas veces, incluyendo modificaciones.
- Carreras disponibles, divididas para cada Facultad.
- Materias correspondientes a cada Carrera.
- Profesores, que pueden ser responsables de una o más asignaturas y con su respectivo cargo en cada una -Profesor Titular o Jefe de Trabajos Prácticos-.
- Horarios de las materias respectivas a cada carrera, para permitir armar los horarios de cada carrera para los distintos ciclos lectivos (años).
- Notificaciones correspondientes a los alumnos, en caso que ocurra algún cambio imprevisto o algún plan reprogramado, como por ejemplo, mesas de consulta, exámenes finales, cambios de horario, etc.
- Resultados de los exámenes de los alumnos, así como también las condiciones del mismo para cada materia –regularidad-.

Por otro lado, el usuario frecuente es el alumno, que tiene accesos exclusivos para ingresar datos específicos, mientras que la mayor interacción es vía móvil, es decir, a través del dispositivo o Smartphone. Entre ellos, el usuario puede:

- Cargar sus datos para darse de alta en el sistema. Esta tarea en usuarios nuevos, permite llenar el formulario para registrarse como nuevo estudiante de la Universidad, con la previa revisión y aprobación del personal correspondiente. Esta tarea alivia al personal de Secretaría la carga de los datos completos de cada alumno, dejando sólo la de aprobación y permitiendo enfocarse en otras tareas.
- Consultar y modificar datos de usuario, para el ingreso al sistema.
- Consultar notas de materias rendidas.
- Consultar horarios de clases a cursar, de acuerdo al año de cursado.
- Verificar horarios que disponen los profesores para las consultas del examen final.
- Realizar las inscripciones a los exámenes finales.
- Recibir notificaciones cargadas por el personal del Área de Secretaría, correspondientes a cualquier cambio imprevisto sobre consultas, exámenes, horarios, etc., y también noticias o publicaciones informativas de interés del alumno.

En cuanto a los webservices, se desarrolla un conjunto de scripts que permiten la conexión entre los datos del sistema administrado por el Área de Secretaría y la aplicación móvil que posee el alumno en su smartphone, por lo tanto, las tareas que ejecuta son las mismas que realiza el usuario, y son, en su mayoría, consultas de “sólo lectura”.

Por otra parte, también es menester mencionar algunos requerimientos no funcionales que no son menos importantes, que son las características generales que el sistema debe cumplir, cómo se adapta a las condiciones que el usuario impone y cómo se comporta ante al mismo. Los requerimientos referidos se describen como:

- **Confiable:** es necesario que la información que se almacena en el sistema permita asignar horarios a los profesores registrados de manera confiable y se garantice la coherencia e integridad de los datos utilizados.
- **Comunicabilidad:** los usuarios ingresan al sistema desde cualquier conexión a internet, ya sea banda ancha, ADSL, GPRS, 3G, etc.
- **Eficiencia:** la aplicación garantiza el buen uso de los recursos que utiliza, como la rápida conexión y respuestas a través de los dispositivos.

- Seguridad: la información se almacena en el sistema y solo puede ser accedido por usuarios registrados.
- Manejo: la aplicación posee interfaces de fácil interpretación, claras para cada función específica y comprensibles para permitir a los usuarios desenvolverse fácilmente.

### 4.3. Arquitectura del sistema

En términos generales, el sistema posee una arquitectura orientada a la web (WOA<sup>20</sup>), del tipo cliente-servidor, utilizando un modelo de capas siguiendo el patrón MVC. Esto requiere que el usuario posea un navegador web para ejecutar la aplicación.

El modelo posee tres capas, o niveles, que contemplan particularmente:

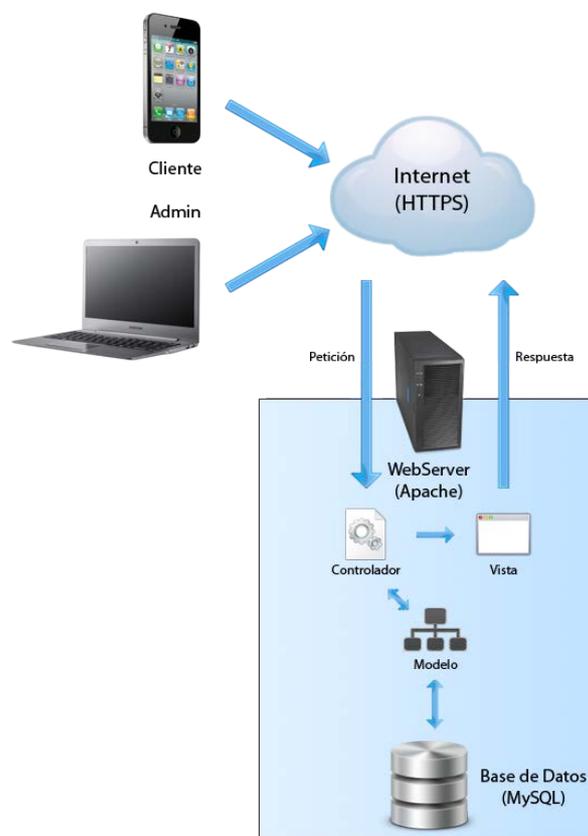
- Primera capa: ubicada en el cliente, se encarga de enviar y recibir la información requerida.
- Segunda capa: ubicada en el servidor, contiene el conjunto de scripts encargados de gestionar y responder las peticiones enviadas por el cliente, aplicando las reglas de negocio impuestas en el sistema.
- Tercera capa: donde se encuentra la base de datos, se almacena la información significativa que gestiona el sistema.

---

<sup>20</sup>*Web-Oriented Architecture*: es un estilo de arquitectura de software que extiende la arquitectura orientada a servicios (SOA) para las aplicaciones basadas en la Web.

La siguiente imagen muestra las diferentes capas, que constituyen el patrón MVC:

**Imagen N° 15:** *Arquitectura basada en la web utilizando MVC.*

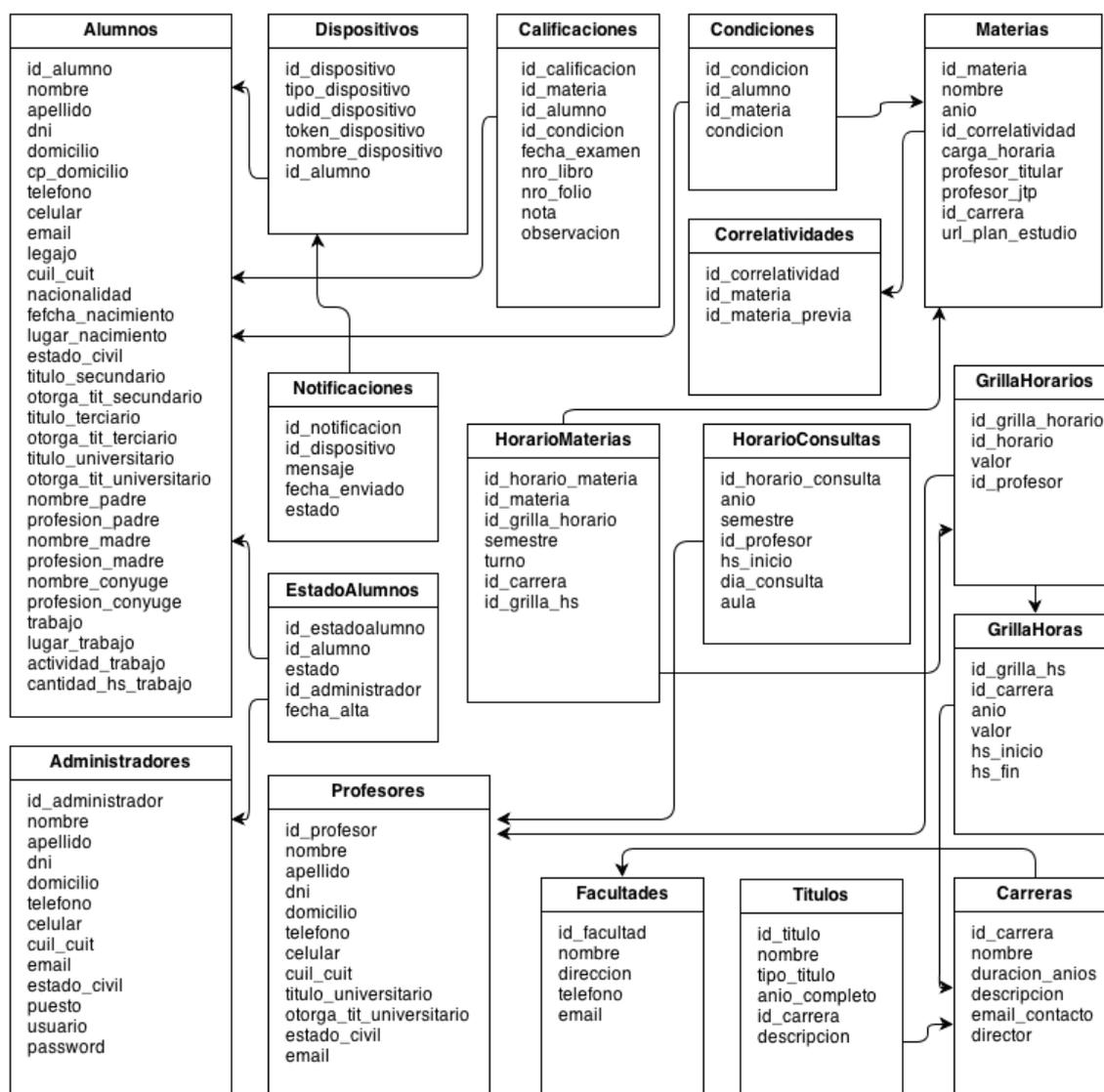


*Fuente: Elaboración propia. Fecha: 2012.*

#### 4.4. Modelo de datos

Teniendo en cuenta los requerimientos que el sistema contiene, una propuesta de modelo de datos que satisfagan las necesidades planteadas es la que se muestra en la siguiente imagen.

**Imagen N° 16:** Modelo de datos del sistema.



*Fuente: Elaboración propia. Fecha: 2013.*

El modelo sugerido puede almacenar información completa del alumno registrado – inscripto- con respecto a sus estudios y calificaciones. Este modelo permite que las consultas

muestren información sobre los datos personales del alumno, con posibilidad de editar los considerados personales.

También contiene información sobre los dispositivos del usuario, donde se condiciona a un máximo de 3 (tres) dispositivos por alumno, como medida de seguridad. Por ejemplo, si el usuario posee un iPhone, un iPod y una iPad, en caso de querer vincular un dispositivo adicional, se debería cancelar uno anterior -se presume que no se tendrían dos celulares al mismo tiempo, en el caso que se quisiera adicionar otro teléfono inteligente-.

Los horarios se representan mediante una grilla que se compone con una letra y un número, haciendo referencia a los días y los módulos respectivamente. Se determinan los horarios de inicio y fin de los módulos y el año al que corresponde el horario. Se disponen horarios de cursado para las materias, de acuerdo a la carrera y año de cursado; así como también horarios de consultas de los diferentes profesores. En estos últimos, se obvian las diferentes materias, ya que las consultas se realizan por el docente en particular, quien abarca todas sus asignaturas dictadas.

Se había hablado anteriormente que el usuario es el encargado de completar toda su información personal a través de un formulario de inscripción. Esto permite eliminar tareas adicionales al personal de Secretaría o encargados, permitiéndoles ocupar el tiempo en otras tareas.

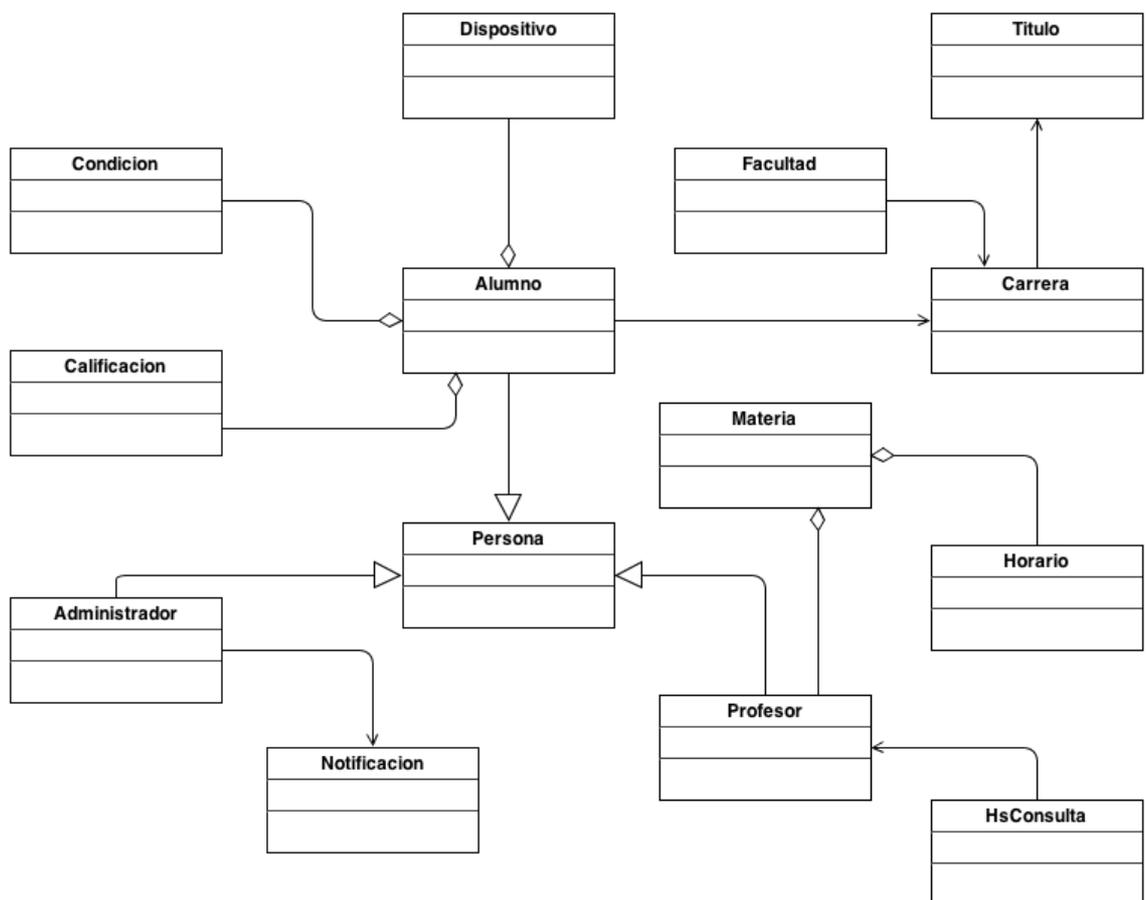
Aunque el usuario o alumno tiene acceso a editar sus datos personales, no tiene la autoridad para darse de alta en una inscripción o de ser habilitado en el sistema sin un previo control, en este caso interviene el administrador, quien controlará que los datos sean correctos y su formulario esté en correcto orden para dar de alta y/o activar al alumno. En caso de aprobación, el alumno podrá comenzar a hacer uso del sistema.

El administrador hace referencia al personal administrativo, ya sea el Área de Secretaría o encargados de tareas afines a la gestión del contenido útil para los alumnos. Las tareas del administrador, principalmente, son las de editar la información sobre las carreras, materias, horarios y la información trascendental para el alumno. Básicamente, el administrador es el encargado de mantener el sistema actualizado y controlado para el correcto funcionamiento por parte de los usuarios.

## 4.5. Diagrama de clases

El modelo de clases básico planteado para el sistema está presentado en la siguiente imagen. El gráfico sólo muestra las clases principales que se utilizarían, teniendo en cuenta el modelo de datos anteriormente descripto.

**Imagen N° 17:** Diagrama de clases básico del sistema.



*Fuente: Elaboración propia. Fecha: 2013.*

De acuerdo al diagrama descripto, existe una generalización de los principales usuarios en una clase *Persona*. De ahí se especializa o hereda en subclases denominadas *Profesor*, *Administrador* y *Alumno*, estas dos últimas son las que interactúan más frecuentemente con el sistema, para actualizar y consultar, respectivamente.

Las clases *Calificación*, *Condición* y *Dispositivo* tienen una relación de agregación respecto a la clase *alumno*, en base a la cardinalidad que se puede presentar. Los alumnos pueden tener varios dispositivos (con un máximo de 3) al igual que contienen condiciones y calificaciones para cada materia.

La clase *Carrera* tiene su respectiva relación con las clases *Título* (respecto a la determinada carrera) y *Facultad* (a la cual pertenece dicha carrera). Del mismo modo ocurre con las clases relacionadas con *Materia*, para la que se agregan profesores asignados a la misma (ej, titular y jefe de trabajos prácticos), mientras que también se posee una agregación de horarios correspondientes a los diferentes turnos de cursado, ya sean de mañana, tarde o noche.

La clase *Notificación* contiene las características de las notificaciones que se generan por el administrador, por lo que son creadas a través del mismo. Las notificaciones tendrán interacción con los dispositivos registrados.

#### **4.6. Servicio de notificaciones de Apple**

El servicio de notificaciones o *Apple Push Notification Service* (APNs) es la herramienta que brinda Apple para generar notificaciones dentro de iOS. Este servicio permite enviar información a los dispositivos que utilizan este Sistema Operativo.

La notificación remota fluye en una dirección. Básicamente, el proveedor compone un paquete de notificación que incluye el token del dispositivo para la aplicación cliente. El proveedor envía la notificación al servicio APN que a su vez empuja la notificación al dispositivo.

Cuando un proveedor se autentifica ante el APNs, envía su notificación al mismo, que identifica la aplicación para la que se está proporcionando los datos. La imagen siguiente ilustra una conexión simplificada.

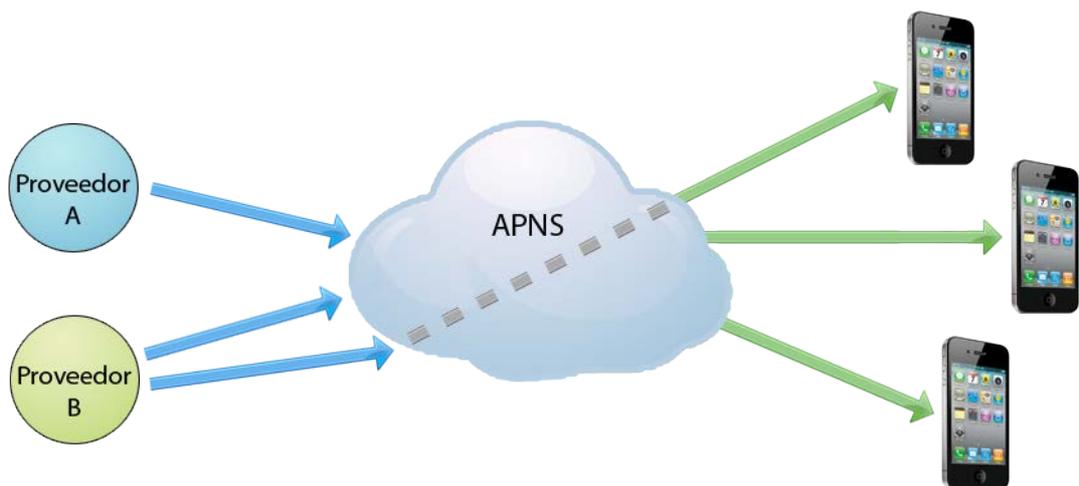
**Imagen N° 18:** Notificación Push desde el proveedor hasta la aplicación cliente.



*Fuente: Apple. Fecha: 2012.*

En general, existen varios proveedores, cada uno haciendo una o más conexiones persistentes a través de estos enlaces. Estos proveedores están enviando notificaciones a través de APN a muchos dispositivos en los que se instalan aplicaciones cliente. Una representación más completa sería la siguiente imagen.

**Imagen N° 18:** Notificaciones Push de varios proveedores a varios dispositivos.



*Fuente: Apple. Fecha: 2012.*

El servicio APN también ofrece a los proveedores información acerca de las notificaciones que no pudieron ser entregadas, por ejemplo, debido a que la aplicación de destino ya no está instalada en ese dispositivo.

#### 4.6.1. Calidad de Servicio (QoS)

El APNs incluye una calidad predeterminada de calidad de servicio (QoS) que consiste en realizar la función *almacenar y reenviar* (store-and-forward).

Si el APNs intenta entregar una notificación pero el dispositivo no está en línea, la notificación se almacena durante un período limitado de tiempo para entregar al dispositivo cuando esté disponible (conectado).

Sólo se almacena una notificación por cada aplicación particular, es decir, la más reciente. Si hay varias notificaciones se envían mientras el dispositivo está en línea y cada nueva notificación hace que la notificación previa a ser desechada. Este comportamiento de mantener sólo la notificación más reciente se conoce como notificaciones fusionadas.

Si el dispositivo permanece fuera de línea por un largo tiempo, las notificaciones que se están almacenando para el mismo se descartan.

#### 4.6.2. Arquitectura de seguridad

Para permitir la comunicación entre un proveedor y un dispositivo, el APNs debe exponer ciertos puntos de entrada a los mismos. Pero para garantizar la seguridad, también debe regular el acceso a estos puntos de entrada. Para ello, APN requiere dos niveles de confianza de los proveedores, los dispositivos y sus comunicaciones. Estos se conocen como la conexión de confianza y el token<sup>21</sup> de confianza (*notification trust* y *token trust*).

---

<sup>21</sup>El *Token* es un conjunto de caracteres encriptados que representan una clave única para identificar un componente en particular.

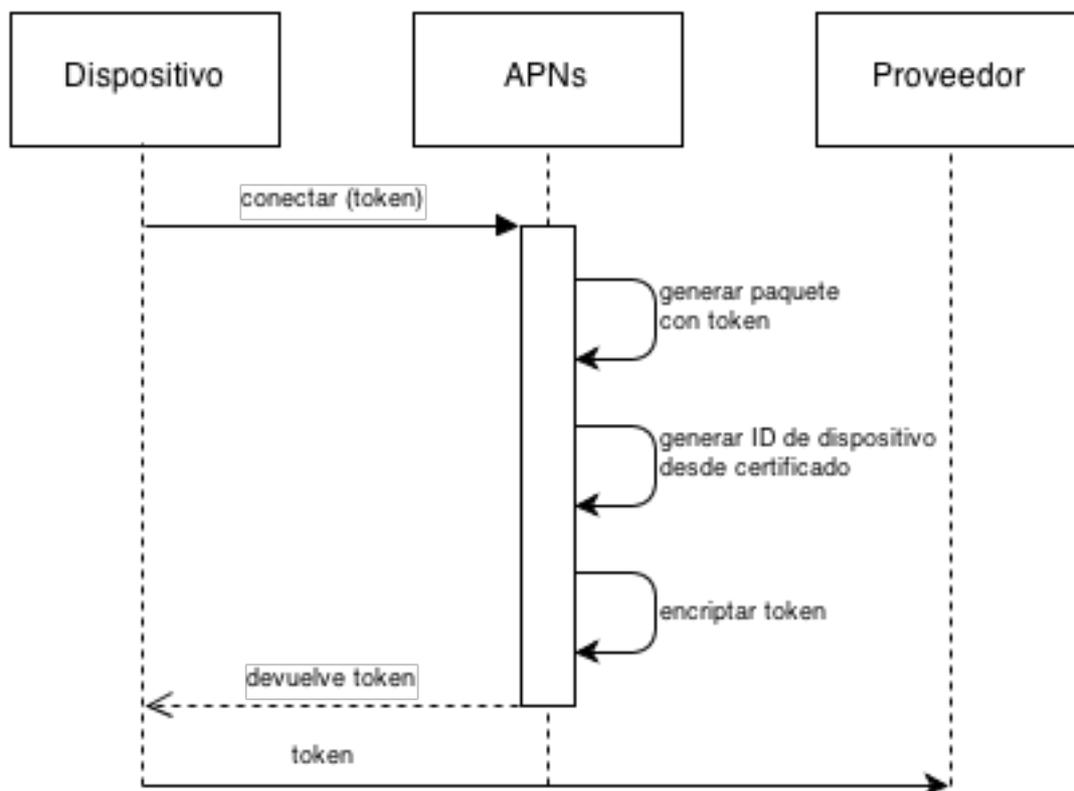
La conexión de confianza establece la certeza de que la conexión APN se realiza con un proveedor autorizado de Apple, quien se compromete a entregar las notificaciones. En el lado del dispositivo, el APNs debe validar que la conexión que se establece es con un dispositivo legítimo.

Después de que el APN ha establecido una conexión de confianza en los puntos de entrada, entonces debe garantizar que se transmiten notificaciones a los únicos puntos finales legítimos. Para ello, se debe validar el enrutamiento de mensajes que viajan a través del transporte, sólo el dispositivo que es el destino previsto debe recibir la notificación.

#### **4.6.3. Generación de token**

La aplicación debe registrarse para recibir notificaciones push, que por lo general lo hace cuando se instala en el dispositivo. El sistema recibe la solicitud de registro de la aplicación, se conecta con la APN, y envía la solicitud. APN genera un token utilizando la información contenida en el certificado del dispositivo único. El token contiene un identificador del dispositivo. A continuación, se cifra el token con una llave y lo devuelve al dispositivo. El equipo regresa el token para la aplicación. Luego, la aplicación debe entregar el token a su proveedor. La imagen muestra el diagrama de secuencia descripto.

*Imagen N° 19: Generación del token.*

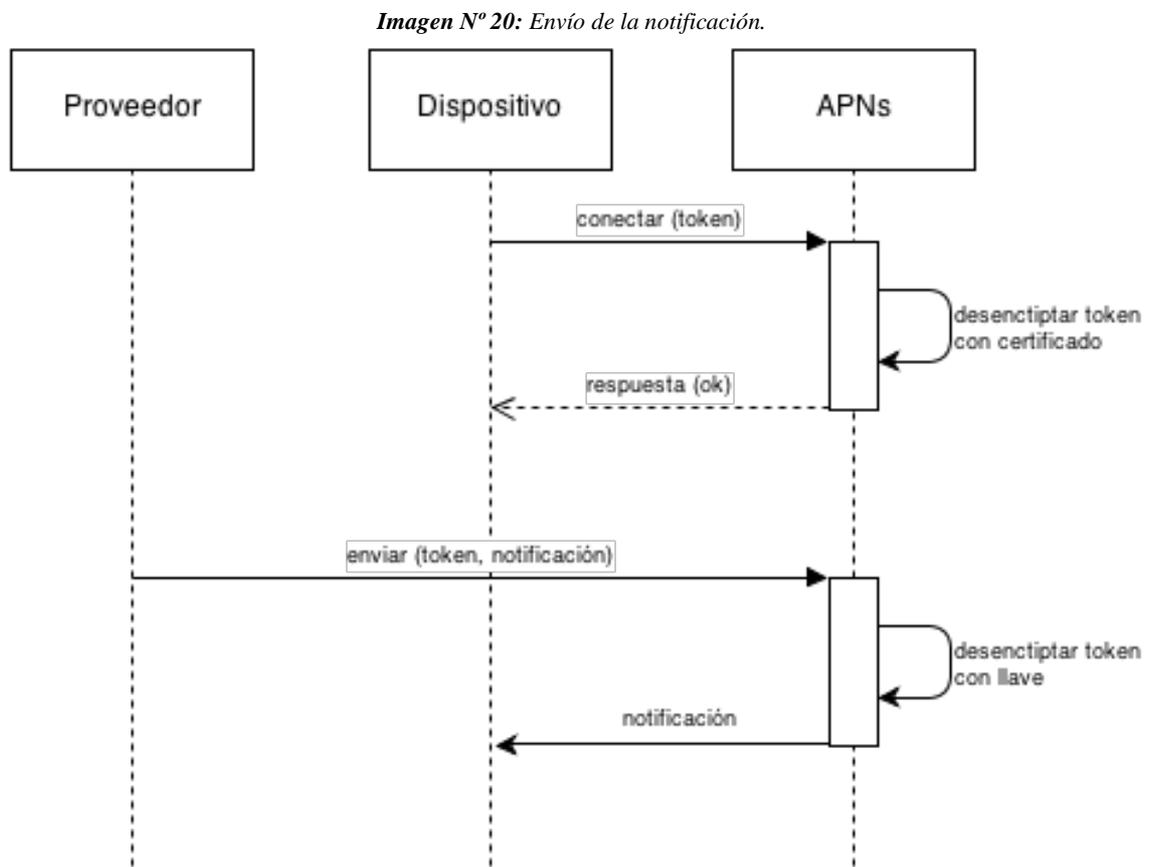


*Fuente: Apple. Fecha: 2012.*

#### 4.6.4. Token de confianza (Token Trust)

Después de que el sistema obtiene el token del dispositivo, debe proporcionar el APNs con ese token cada vez que se conecte. La APN descifra la señal del dispositivo y valida que el token se genera para el dispositivo específico. Para validar, el APNs se asegura de que el ID del dispositivo que se encuentra en el token coincida con el ID del dispositivo especificado en el certificado de dispositivo.

Cada notificación enviada por el proveedor al APN debe tener el token del dispositivo obtenida desde la aplicación. La APN descifra el token usando la llave, garantizando así que la notificación sea válida. Luego, utiliza el ID del dispositivo que se encuentra en el token para determinar el dispositivo destino al cual se le enviará la notificación. La siguiente imagen muestra el diagrama de secuencia.



*Fuente: Apple. Fecha: 2012.*

## 4.7. Casos de Uso

Los casos de uso básicamente representan las interacciones que algo o alguien tiene con el sistema. Estas entidades externas son conocidas como *actores*. Los actores tienen una acción específica con los diferentes módulos o servicios que el sistema provee. De esta manera, los casos de uso permiten describir el comportamiento del sistema, desde el punto de vista del actor, por ejemplo, el usuario.

### 4.7.1. Diagramas de casos de uso

Los diagramas de casos permiten documentar de manera gráfica el comportamiento del sistema. Generalmente se representa las funciones que el sistema puede ejecutar.

Al ser diagramas de interpretación sencilla, esta herramienta se hace muy útil para la comunicación con el cliente, ya que permite mostrar de una manera simple las características que el sistema contiene.

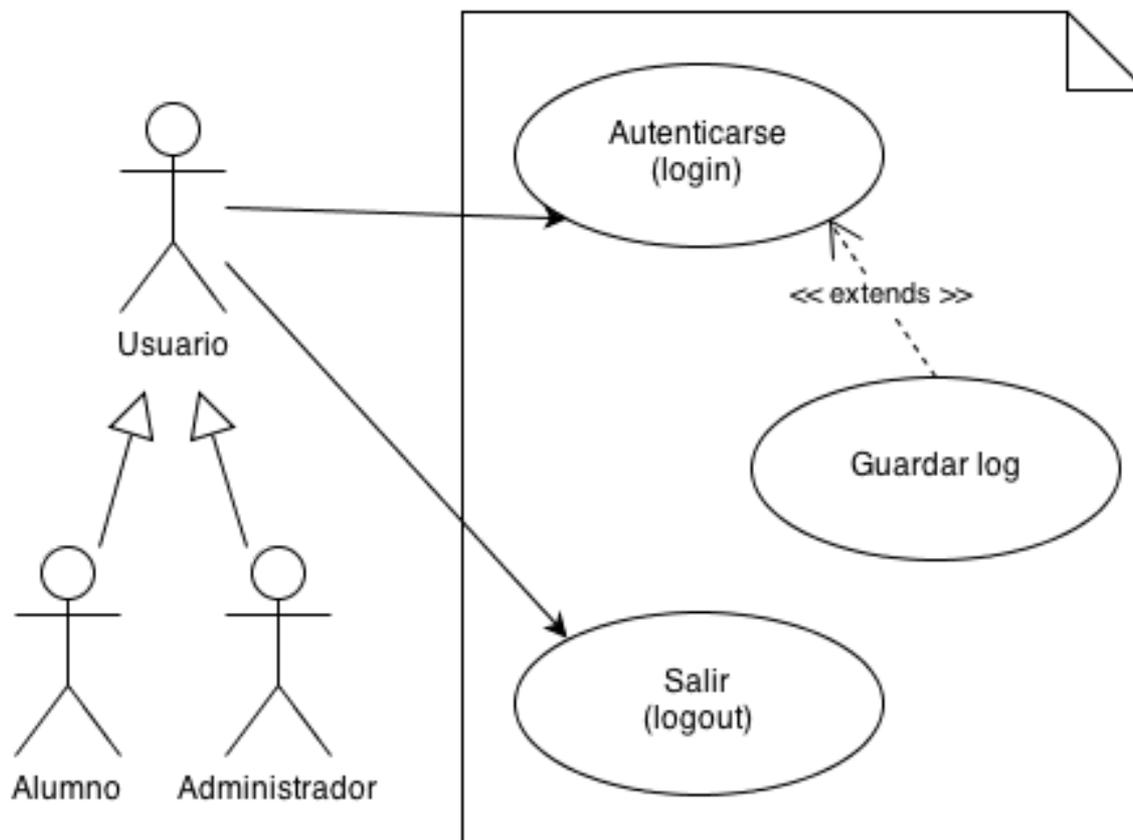
Básicamente, los diagramas de casos de uso contienen los siguientes componentes:

- *Actor*: toda entidad externa que interactúa directamente con el sistema. Puede ser un usuario humano o incluso otro sistema informático. Se representan con un muñeco de palos.
- *Caso de uso*: las tareas que pueden llevarse a cabo con el sistema. Se representan con un ovalo, especificando su funcionamiento escrito dentro del mismo.
- *Asociaciones*: es el enlace que determina que existe la interacción directa entre el actor y el caso de uso. Se grafica con una línea entre ambos componentes.
- *Escenario*: es el flujo o comportamiento específico que se desea representar en el diagrama, es decir, la acción particular y la/s respuesta/s que el sistema otorga ante esta acción.

A continuación se describirán algunos de los diagramas de casos de uso más importantes del sistema, con sus respectivas representaciones.

#### 4.7.1.1. Caso de Uso Autenticarse

Imagen N° 21: Diagrama de Caso de Uso: Autenticar.



*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.1.1. Descripción del Caso de Uso

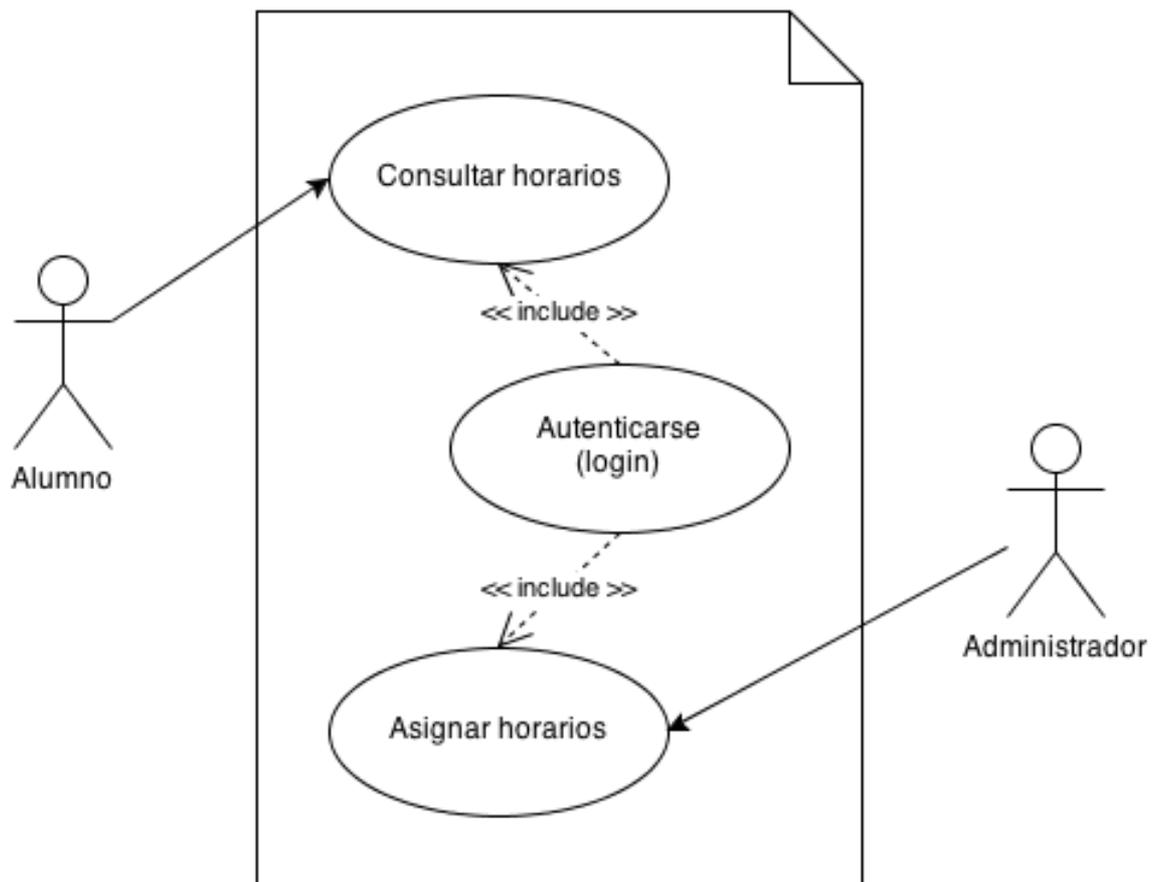
*Tabla N° 3: Descripción de Caso de Uso: Autenticarse.*

<b>Nombre:</b> Autenticarse (login)	
<b>Actores:</b> Usuario (Alumno, Administrador)	
<b>Objetivo:</b> Ingresar al sistema de manera segura.	
Acción del Actor	Acción del Sistema
<i>Flujo principal</i>	
1. El CU comienza cuando el usuario ingresa su usuario y contraseña para autenticarse.	2. El sistema realiza una búsqueda del usuario en su base de datos.
	3. El CU finaliza cuando se inicia una sesión para el usuario y se ingresa al sistema.
	4. <<extends>> Se guarda un registro (log) con los datos del usuario, fecha de ingreso y dispositivo, entre otros.
<i>Flujo alternativo</i>	
	3. El sistema devuelve un error de autenticación incorrecta.

*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.2. Caso de Uso Consultar horarios y Asignar horarios

Imagen N° 22: Diagrama de Caso de Uso: Consultar y Asignar horarios.



*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.2.1. Descripción del Caso de Uso

*Tabla N° 4: Descripción de Caso de Uso: Consultar horarios.*

<b>Nombre:</b> Consultar horarios	
<b>Actores:</b> Alumno	
<b>Objetivo:</b> Consultar horarios de clase y consultas de profesores.	
Acción del Actor	Acción del Sistema
<i>Flujo principal</i>	
1. << include >> CU: Autenticarse.	
2. El alumno selecciona la opción de ‘consultar horarios’.	
3. El alumno elige la opción ‘materias’.	
4. El usuario elige la materia que desea consultar.	
	5. El CU finaliza cuando el sistema busca la materia en los horarios y la muestra.
<i>Flujo alternativo 1</i>	
3. El alumno selecciona la opción de consultar semestre.	
	4. El sistema busca las materias del semestre.
	5. El CU finaliza cuando el sistema arma los horarios del semestre y los muestra en un calendario.
<i>Flujo alternativo 2</i>	
3. El alumno elige la opción de ‘consultas de profesores’	
4. El alumno elige el profesor a consultar.	
	5. El CU finaliza cuando el sistema busca el horario de consulta del profesor y la muestra.

*Fuente: Elaboración propia. Fecha: 2013.*

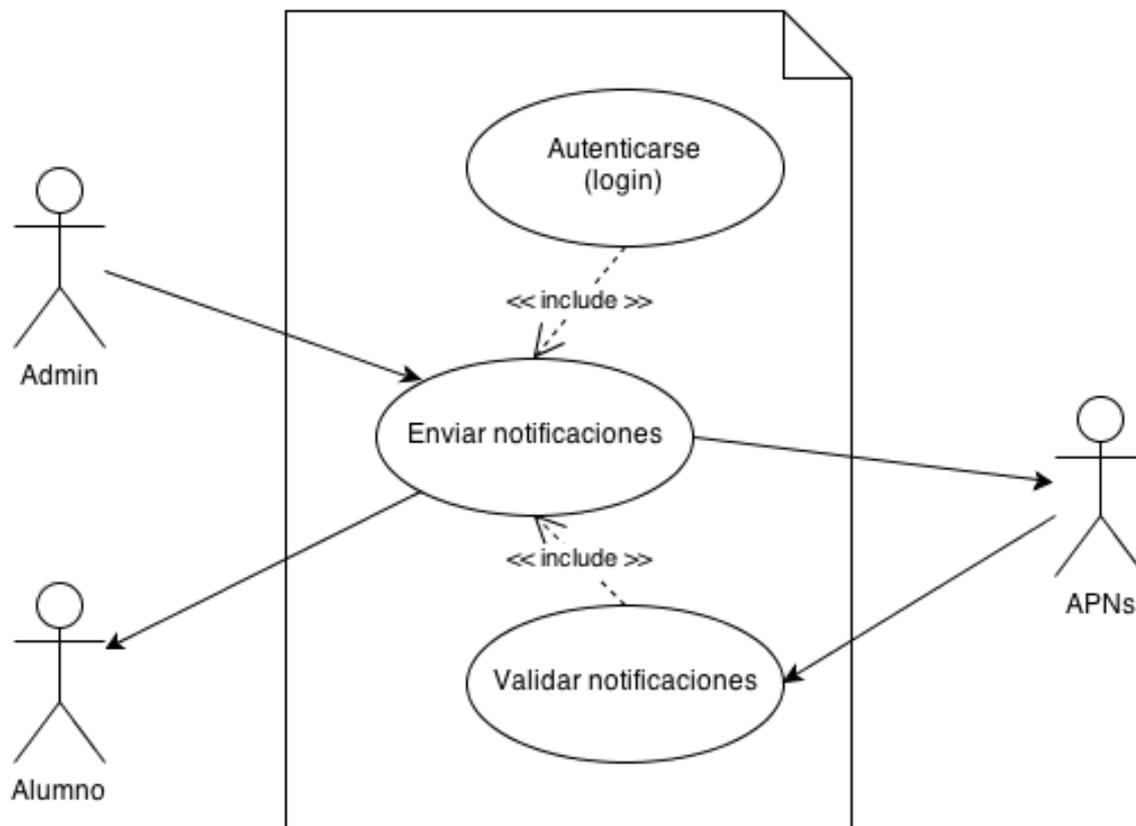
*Tabla N° 5: Descripción de Caso de Uso: Asignar horarios.*

<b>Nombre:</b> Asignar horarios	
<b>Actores:</b> Administrador	
<b>Objetivo:</b> Asignar los horarios de clase y consultas de profesores.	
<b>Acción del Actor</b>	<b>Acción del Sistema</b>
<i>Flujo principal</i>	
1. << incluye >> CU: Autenticarse.	
2. El administrador ingresa en la opción 'cargar horarios'.	
3. El administrador elige la opción 'materias'.	
4. El administrador selecciona los datos que desea cargar, la carrera, año, semestre y turno.	5. El sistema proporciona un calendario para ubicar las materias.
6. El administrador ubica las materias en el calendario.	7. El CU finaliza cuando el sistema guarda la información en su base de datos e informa que el proceso se ha realizado satisfactoriamente.
<i>Flujo alternativo 1</i>	
3. El administrador elige la opción de 'profesores'.	
4. El administrador elige un profesor y llena su disponibilidad horaria.	5. El CU finaliza cuando el sistema guarda la información en su base de datos e informa que el proceso se ha realizado satisfactoriamente.

*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.3. Caso de Uso Enviar notificaciones

Imagen N° 23: Diagrama de Caso de Uso: Enviar notificaciones.



*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.3.1. Descripción del Caso de Uso

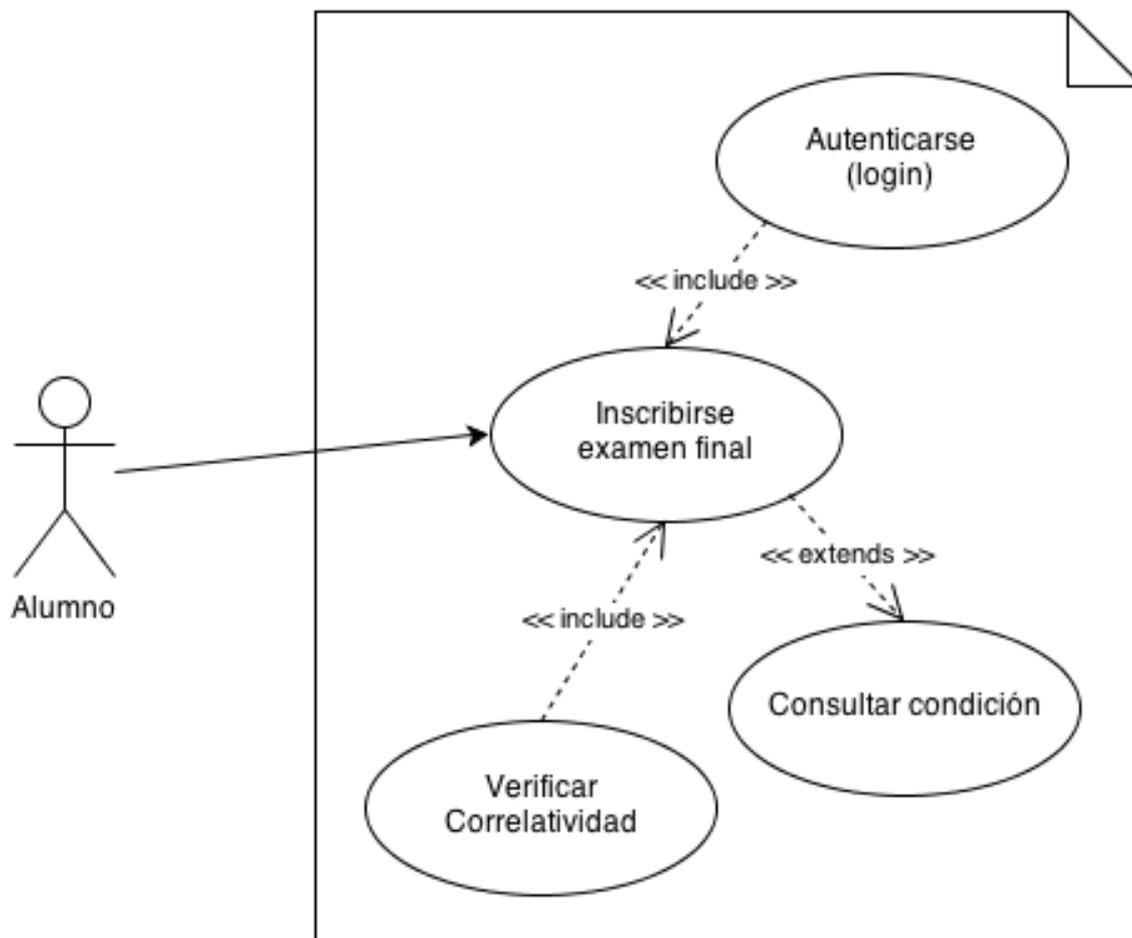
*Tabla N° 6: Descripción de Caso de Uso: Enviar notificaciones.*

<b>Nombre:</b> Enviar notificaciones	
<b>Actores:</b> Administrador, Alumno, APNs (Apple Push Notification Services)	
<b>Objetivo:</b> Configurar el envío de notificaciones para los usuarios (Alumnos)	
Acción del Actor	Acción del Sistema
<i>Flujo principal</i>	
1. <<include>> CU: Autenticarse.	
2. El administrador guarda la notificación que desea informar.	3. El sistema guarda en su base de datos la notificación y avisa al APNs.
	4. El sistema envía un aviso al servicio APN.
5. El APNs valida la notificación enviada y devuelve la difusión.	<<include>>CU: Verificar notificaciones
	6. El CU finaliza cuando el sistema envía la notificación al usuario final.

*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.4. Caso de Uso Inscribirse a examen final

Imagen N° 24: Diagrama de Caso de Uso: Inscribirse a examen final.



*Fuente: Elaboración propia. Fecha: 2013.*

#### 4.7.1.4.1. Descripción del Caso de Uso

*Tabla N° 7: Descripción de Caso de Uso: Inscribir a examen final.*

<b>Nombre:</b> Inscribirse a examen final	
<b>Actores:</b> alumno	
<b>Objetivo:</b> Realiza la inscripción al examen final de una materia.	
<b>Acción del Actor</b>	<b>Acción del Sistema</b>
<i>Flujo principal</i>	
1. <<include>> CU: Autenticarse	
2. El alumno ingresa al sistema para inscribirse a un examen final.	
3. El alumno elige un año y una materia específica.	4. <<incl>> CU: Verificar correlatividad. El sistema verifica la correlatividad de la materia para determinar si el alumno está apto para rendir.
	5. El sistema realiza la inscripción al examen final para la materia.
	6. El CU finaliza cuando el sistema imprime un comprobante con los datos del examen y anota el compromiso en la agenda del dispositivo.
<i>Flujo alternativo 1</i>	
2. <<extends>> CU: Consultar condición El alumno ingresa la opción de consultar condición.	
3. El alumno elige la materia a consultar	4. El sistema busca en la base de datos la regularidad dela materia.
	5. El CU finaliza cuando el sistema devuelve la condición de la materia actual.

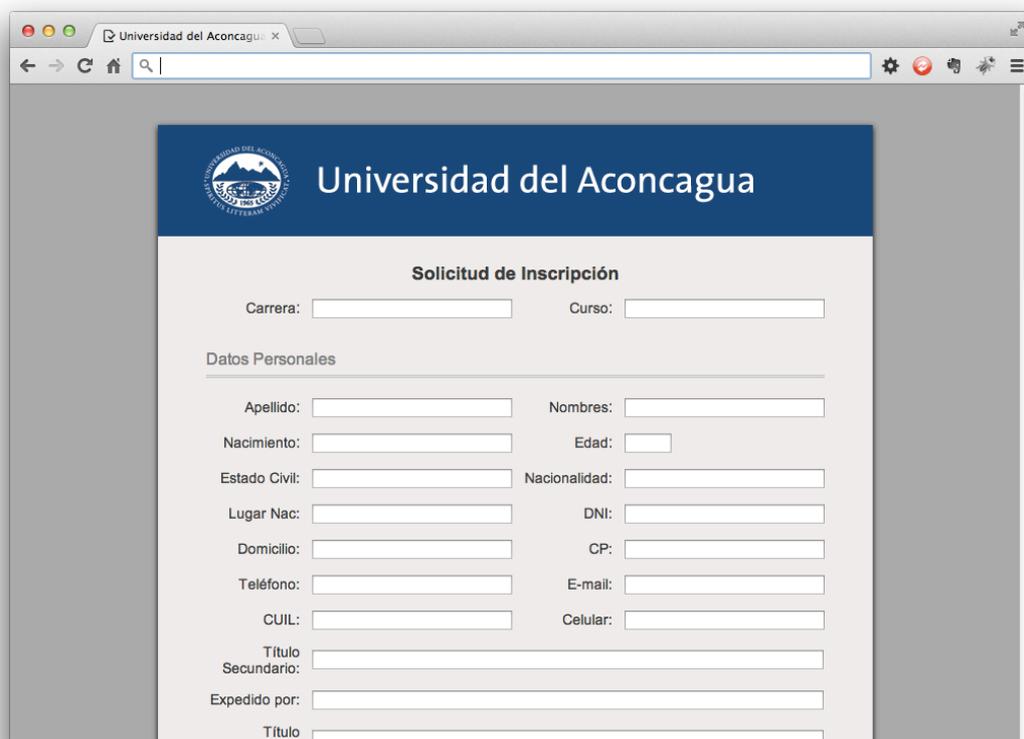
*Fuente: Elaboración propia. Fecha: 2013.*

## 4.8. Diseño de interfaces

### 4.8.1. Screenshots

Continuando con la presentación de interfaces del sistema, esta vez enfocado en la plataforma web, las siguientes imágenes muestran el diseño propuesto para la aplicación utilizada sobre un navegador web estándar.

*Imagen N° 25: Diseño de formulario de inscripción.*

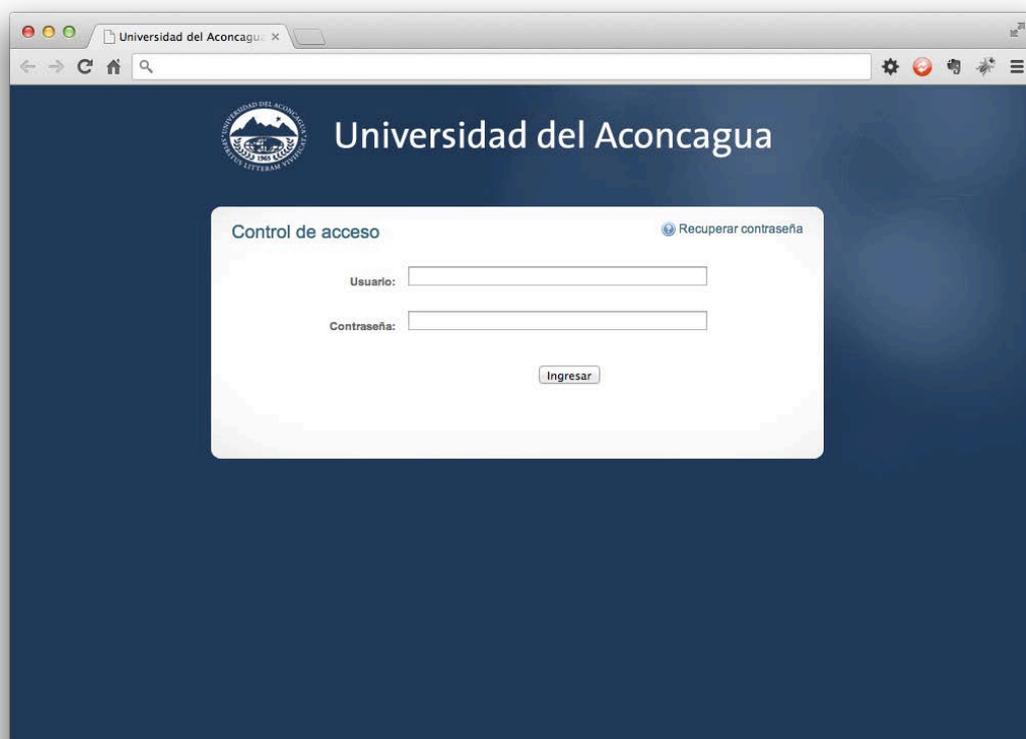


The image shows a web browser window with the URL 'Universidad del Aconcagua'. The page features a blue header with the university's logo and name. Below the header is a form titled 'Solicitud de Inscripción'. The form includes fields for 'Carrera' and 'Curso'. A section titled 'Datos Personales' contains fields for 'Apellido', 'Nombres', 'Nacimiento', 'Edad', 'Estado Civil', 'Nacionalidad', 'Lugar Nac.', 'DNI', 'Domicilio', 'CP', 'Teléfono', 'E-mail', 'CUIL', and 'Celular'. At the bottom, there are fields for 'Titulo Secundario', 'Expedido por', and 'Titulo'.

*Fuente: Elaboración propia. Fecha: 2013.*

El formulario de inscripción permite al alumno realizar este trámite de manera online, simplificando la tarea, tanto para él como para el personal administrativo.

*Imagen N° 26: Diseño de ingreso (login).*

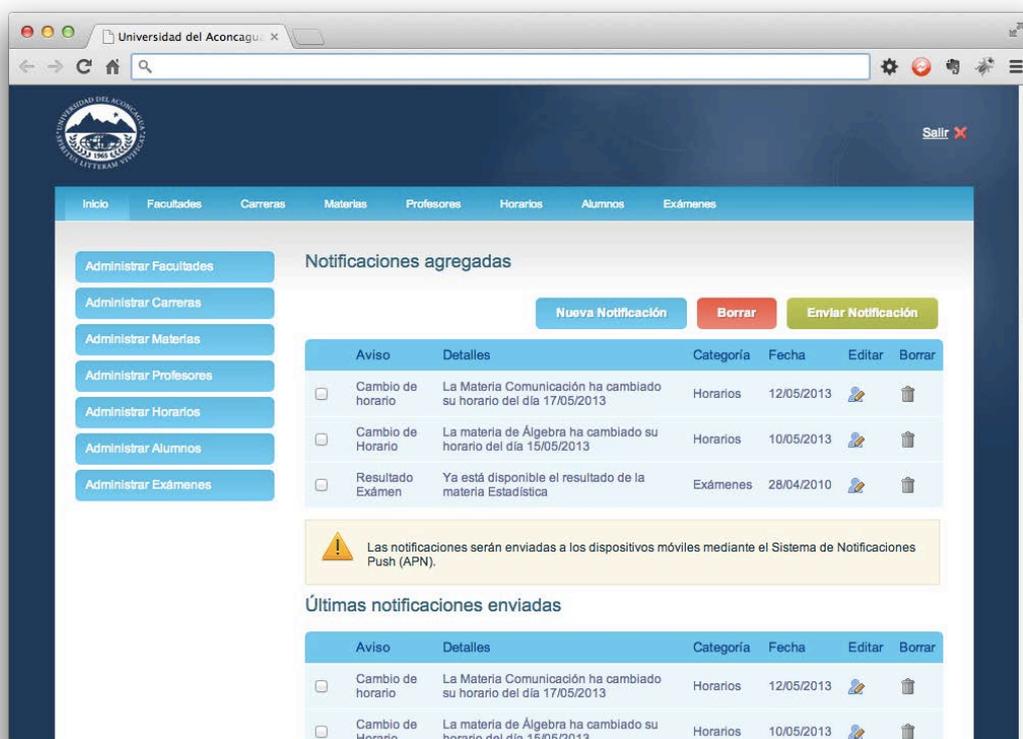


The image shows a web browser window displaying the login page for Universidad del Aconcagua. The browser's address bar shows the URL "Universidad del Aconcagua". The page has a dark blue background. At the top left is the university's logo, and to its right is the text "Universidad del Aconcagua". Below this, there is a white box titled "Control de acceso" with a link for "Recuperar contraseña". The login form includes two input fields: "Usuario:" and "Contraseña:", followed by an "Ingresar" button.

*Fuente: Elaboración propia. Fecha: 2013.*

La pantalla de login permite el ingreso del administrador al sistema.

*Imagen N° 27: Diseño de inicio de Administrador.*



*Fuente: Elaboración propia. Fecha: 2013.*

El inicio del administrador muestra las opciones posibles para el personal, accediendo principalmente al panel de notificaciones. Permite agregar notificaciones y colocarlas en una lista para poder ser enviadas. Otra lista muestra las notificaciones enviadas recientemente.

*Imagen N° 28: Diseño de formulario de notificaciones.*

Universidad del Aconcagua

Salir

Inicio Facultades Carreras Materias Profesores Horarios Alumnos Exámenes

Administrar Facultades  
Administrar Carreras  
Administrar Materias  
Administrar Profesores  
Administrar Horarios  
Administrar Alumnos  
Administrar Exámenes

Nueva Notificación

Titulo:

Categoría:

Fecha:

Facultades:

Carreras:

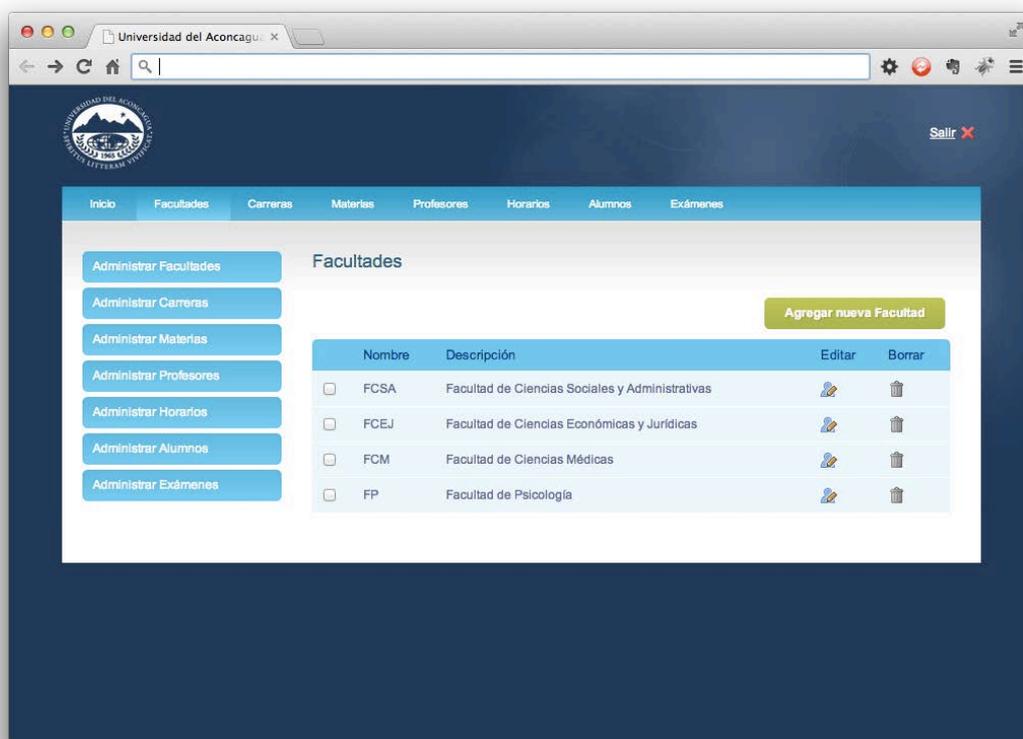
Detalle:

Agregar

*Fuente: Elaboración propia. Fecha: 2013.*

El formulario de notificaciones es muy sencillo, se detalla los datos del mensaje y se añade el destino del mensaje, es decir, la facultad y la carrera en particular. Luego, desde el panel de inicio se pueden enviar las notificaciones cuando se considera necesario.

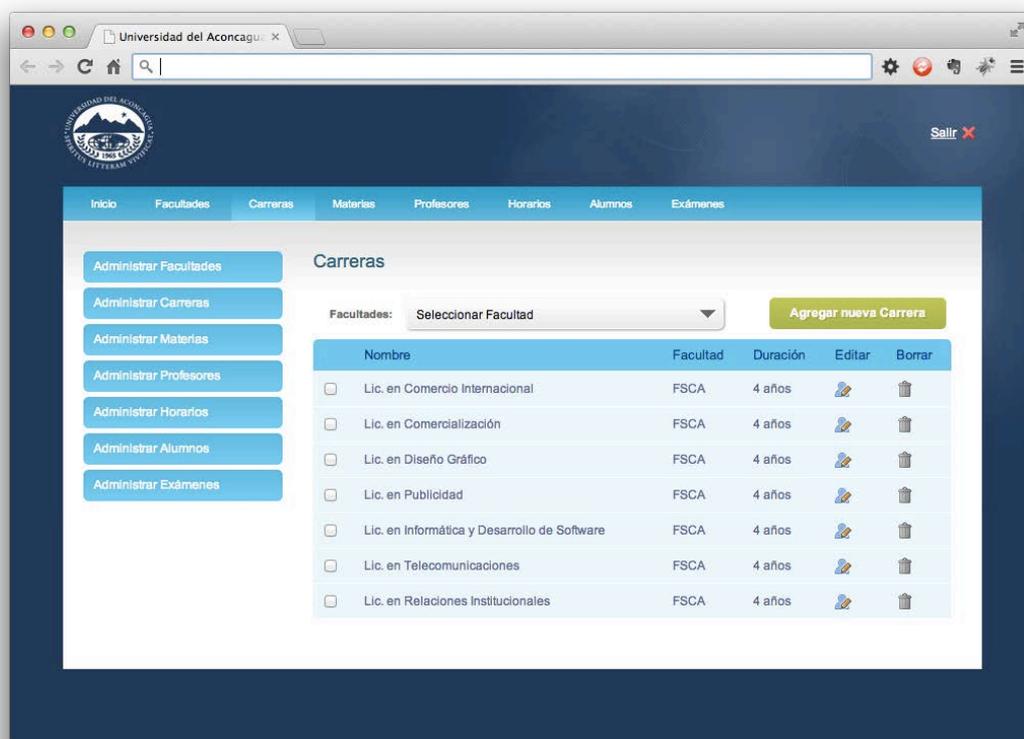
*Imagen N°29: Diseño de listado de Facultades.*



*Fuente: Elaboración propia. Fecha: 2013.*

El listado muestra las facultades disponibles o existentes en la Universidad. También se puede acceder a la edición de las mismas.

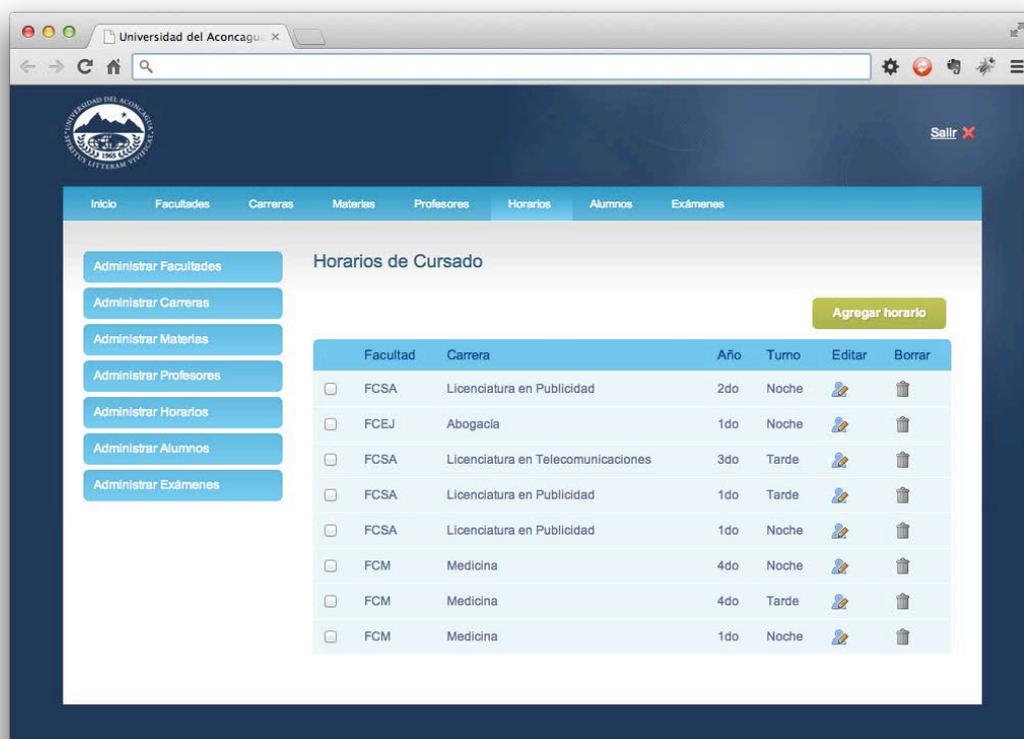
*Imagen N° 30: Diseño de listado de Carreras.*



*Fuente: Elaboración propia. Fecha: 2013.*

El listado de Carreras es similar al anterior, permitiendo filtrar por Facultades. Las acciones son las mismas.

*Imagen N° 31: Diseño de listado de Horarios.*



*Fuente: Elaboración propia. Fecha: 2013.*

La sección de horarios muestra el listado de los horarios de cursado subidos. Desde esta pantalla se accede a generar una nueva grilla para llenar los horarios de cursado de las materias para cada carrera.

*Imagen N° 32: Diseño de Agregar Horario de Carrera.*

The screenshot shows a web browser window with the URL 'Universidad del Aconcagua'. The page features a navigation menu with options: Inicio, Facultades, Carreras, Materias, Profesores, Horarios, Alumnos, and Exámenes. A sidebar on the left contains buttons for: Administrar Facultades, Administrar Carreras, Administrar Materias, Administrar Profesores, Administrar Horarios, Administrar Alumnos, and Administrar Exámenes. The main content area is titled 'Horarios de Cursado' and includes filters for Facultad, Carrera, Año, and Turno. There are 'Agregar' and 'Borrar' buttons. Below the filters is a table with columns for LUNES, MARTES, MIÉRCOLES, JUEVES, and VIERNES. The rows represent time slots: 17:20-18:05, 18:10-18:55, 19:00-19:45, 19:50-20:35, 20:40-21:25, and 21:30-22:15. Each cell in the table contains an empty input box for scheduling.

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
17:20 18:05	<input type="text"/>				
18:10 18:55	<input type="text"/>				
19:00 19:45	<input type="text"/>				
19:50 20:35	<input type="text"/>				
20:40 21:25	<input type="text"/>				
21:30 22:15	<input type="text"/>				

*Fuente: Elaboración propia. Fecha: 2013.*

Esta grilla se completa de acuerdo a los datos específicos de cada carrera. Se completan como una grilla de acuerdo a los días hábiles de cursado y sus respectivos horarios, siguiendo el turno especificado.

En cuanto a las pantallas de la aplicación del dispositivo móvil, las imágenes de las más utilizadas se muestran a continuación:

*Imagen N°33: Diseño de ingreso a la app (login).*



*Fuente: Elaboración propia. Fecha: 2013.*

Luego de la pantalla de bienvenida, las seguidas son las de ingreso al sistema, a través de la verificación de usuario, o login. Las vistas son sencillas e intuitivas, donde se solicita al usuario su número de legajo y su DNI, seguidamente y luego de una búsqueda en la base de datos, la contraseña del usuario o alumno.

*Imagen N°34: Diseño de inicio y listado de progreso.*



*Fuente: Elaboración propia. Fecha: 2013.*

El inicio o principal de la aplicación consta de varias opciones que permiten al usuario realizar diferentes tareas, tales como revisar su progreso de la carrera, consultar horarios, revisar notificaciones enviadas, inscribirse a un examen, revisar notas de exámenes anteriores, comunicarse con la respectiva facultad por vía telefónica, entre otras.

El progreso muestra una lista que divide la información por año, o bien, por la carrera completa.

Imagen N° 35: Diseño de inscripción a examen.



*Fuente: Elaboración propia. Fecha: 2013.*

La inscripción a un examen se realiza seleccionando de una lista dividida por años y materias no rendidas aún, y se visualiza el detalle de la materia a rendir, para solicitar la inscripción a la materia deseada.

*Imagen N° 36: Diseño de listado de Horarios.*



*Fuente: Elaboración propia. Fecha: 2013.*

Otra de las vistas importantes son la de consulta a horarios, donde se puede acceder a los días de la semana y en detalle se muestran en una lista, con los horarios y el profesor a cargo.

Se puede notar que las pantallas son la representación los bocetos realizados en los mockup diseñados anteriormente, con los componentes elegidos y el flujo de pantallas que seguirán para una navegación intuitiva y sencilla.

## 4.9. Opinión personal

Se encontró que la plataforma XCode es sencilla para realizar el desarrollo del sistema si se utiliza la mayoría de las herramientas que brinda el SDK. Sin embargo, personalmente se prefiere implementar algunas funcionalidades directamente por código, de manera que se tenga el control sobre lo que se realiza (a veces la ayuda de la herramienta puede ser un problema en un desarrollo complejo).

En cuanto a las revisiones por parte del equipo de Apple, las mismas suelen durar alrededor de 10 días aproximadamente. Los resultados pueden sorprender bastante debido a que las revisiones son manuales y realizadas por seres humanos. Éstas difieren en sus criterios de control, y si bien se sigue una normativa, muchas veces se pasan por alto errores que deberían ser rechazados y otras veces no se aceptan desarrollos que respetan las normas, tal vez por incompreensión por parte del revisor.

Mediante la experiencia personal, se ha observado que las versiones de iOS han resultado ser incompatibles entre sí, afectando a la mayoría de las aplicaciones desarrolladas. El sistema operativo pretende mejorar su calidad en cada versión, modificando algunas funcionalidades que faciliten su implementación. Sin embargo, esto hace que el desarrollo actual falle en las distintas versiones del SO. Por ello, hay que estar atentos: descargar el nuevo iOS y verificar que la aplicación siga funcionando correctamente, ya que si no lo hace hay que revisar el problema y subir una nueva actualización. A favor de esto se puede exigir al grupo de Apple una revisión expedita explicando el problema específico, y en estos casos la revisión puede tardar mucho menos.

Una recomendación a tener en cuenta es la expansión del sistema a otras plataformas, por ejemplo Android, quien ocupa actualmente una parte muy importante del mercado. Esto requiere conocer el lenguaje y las herramientas que brinda Google para el desarrollo de dichas plataformas.

Si bien el desarrollo en iOS puede ser algo tedioso para el programador por las normativas de Apple, sus revisiones y su poca compatibilidad entre sus versiones, todo esto tiene la finalidad de garantizar que el producto sea confiable para el usuario final.

## 5. Conclusión

De acuerdo al presente trabajo, es necesario destacar los puntos que favorecen el desarrollo planteado:

- Las aplicaciones móviles para smartphones a menudo trabajan de manera más directa de acuerdo a las demandas que el usuario reclama, ya que tienen un diseño sencillo. Por otra parte son mas rápidas en tiempos de respuesta, ya que contemplan un flujo leve de información, permitiendo que el gasto en el uso de datos sea menor que un sitio web convencional, incluso para web apps adaptadas o con diseño responsivo.
- La implementación de una aplicación móvil en la Universidad, permite que el alumno contenga el material administrativo que necesite y pueda realizar operaciones académicas y consultas específicas, recibir noticias importantes de manera instantánea, y todo esto justo en su bolsillo, sin necesidad de establecerse en un solo lugar o asistir al establecimiento universitario.

## 6. Fuentes

- AMAZON (2005). REST vs SOAP. Recuperado el 10 de Marzo de 2013, de [http://aws.typepad.com/aws/2005/09/rest\\_vs\\_soap.html](http://aws.typepad.com/aws/2005/09/rest_vs_soap.html)
- APPLE Inc. (2012). Apple Push Service. Recuperado el 21 de Marzo de 2012, de <http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html>,  
[http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html#//apple\\_ref/doc/uid/TP40008194-CH100-SW9](http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html#//apple_ref/doc/uid/TP40008194-CH100-SW9),  
<https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ProvisioningDevelopment/ProvisioningDevelopment.html>
- APPLE Inc. (2012). Appstore Approval Process. Recuperado el 13 de Diciembre de 2012, de <https://developer.apple.com/appstore/resources/approval/index.html>
- APPLE Inc. (2012). Appstore Review Guidelines. Recuperado el 13 de Diciembre de 2012, de <https://developer.apple.com/appstore/resources/approval/guidelines.html>
- APPLE Inc. (2012). Introduction to Objective-C. Recuperado el 23 de Marzo de 2012, de <https://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
- APPLE Inc. (2012). The Foundation Framework. Recuperado el 13 de Diciembre de 2012, de [http://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/ObjC\\_classic/\\_index.html](http://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/ObjC_classic/_index.html)
- APPLE Inc. (2013). Model-View-Controller. Recuperado el 16 de Agosto de 2012, de <https://developer.apple.com/library/mac/#documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- AXTERCIYO (2011). Evolución: Del Móvil a la Tablet. Recuperado el 7 de Marzo de 2012, de <http://debuenamano.wordpress.com/2011/12/22/evolucion-del-movil-a-la-tablet/>
- AYALA, Dietrich - NICHOL, Scott (2006). NuSOAP - SOAP Toolkit for PHP. Recuperado el 2013, de <http://sourceforge.net/projects/nusoup/>
- BOOCH, Grady - JACOBSON, Ivar - RUMBAUGH, James. El Lenguaje Unificado del Modelado - Addison Wesley - 2000.

- BREA, Orlando Fabián (2003). Servicios Web con PHP (NuSOAP). Recuperado el 23 de Mayo de 2013, de <http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>
- BREA, Orlando Fabián (2005). WSDL y UDDI. Recuperado el 6 de Agosto de 2012, de <http://www.desarrolloweb.com/articulos/1857.php>
- BREÑA MORAL, Juan Antonio (2001). ¿Qué son los Web Services?. Recuperado el 11 de Agosto de 2012, de [http://www.webtaller.com/maletin/articulos/que\\_son\\_web\\_services.php](http://www.webtaller.com/maletin/articulos/que_son_web_services.php)
- BRITTENHAN, Peter - CURBERA, Francisco (2001). Understanding WSDL in a UDDI registry. Recuperado el 5 de Agosto de 2012, de <http://www.ibm.com/developerworks/webservices/library/ws-wsdl/>
- CERIA, Santiago (2002). Casos de Uso. Un método práctico para Explorar Requerimientos. Recuperado el 23 de Agosto de 2013, de [http://www-2.dc.uba.ar/materias/isoft1/2001\\_2/apuntes/CasosDeUso.pdf](http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf)
- CISCO MICROSYSTEMS (2012). Recuperado el 21 de Abril de 2013, de <http://www.cisco.com/en/US/solutions/ns341/ns525/ns537/ns705/ns1120/2012-CCWTR-Chapter1-Global-Results.pdf>
- DRAW.IO (2013). Draw.io: Diagramas Online. Recuperado el 7 de Julio de 2013, de <https://www.draw.io/>
- FALCÓN, Cassandra D. (2010). Microprocesadores en smartphones -Evolución multinúcleos. Recuperado el 11 de Agosto de 2012, de <http://www.iuma.ulpgc.es/~nunez/clases-micros-para-com/mpc1011-trabajos/mpc1011-Cassandra-Microprocesadores%20en%20SMARTPHONES.pdf>
- FOSTER, Garland (2002). Tutorial sobre web services: SOAP. Recuperado el 11 de Agosto de 2012, de <http://www.malditainternet.com/2002/01/03/tutorial-sobre-web-services-soap/>
- GOOGLE (2013). Comenzar a utilizar SSL (https:). Seguridad y Autenticación (SSL y HTTPS). Recuperado el 1 de Mayo de 2013, de <http://support.google.com/adwords/answer/2580401?hl=es-419>
- HAIDER, Daniel (2007). Martin Cooper, inventor del celular. Recuperado el 17 de Junio de 2012, de <http://diariodeunteleco.wordpress.com/2007/05/22/martin-cooper-y-la-primera-llamada-desde-un-movil/>
- HARRISON, Keith (2011). iTunes Connect App Status Update. Recuperado el 14 de Diciembre de 2012, de <http://useyourloaf.com/blog/2011/08/29/itunes-connect-app-status-update.html>
- IPSOS MEDIATECH (2013). Our Mobile Planet. Recuperado el 19 de Marzo de 2013, de <http://www.thinkwithgoogle.com/mobileplanet/es/>

- IPSOS MEDIATECT (2013). Recuperado el 19 de Marzo de 2013, de <http://services.google.com/fh/files/misc/omp-2013-ar-local.pdf>
- LOARTE, Vanesa P. (2008). Evolución de los teléfonos celulares y su precio. Recuperado el 25 de Junio de 2012, de <http://www.monografias.com/trabajos51/evolucion-celulares/evolucion-celulares2.shtml>
- LÓPEZ BRAGUE, Carlos. (2011) Evolución de los Smartphones. Recuperado el 16 de Junio de 2012, de <http://es.scribd.com/doc/55081015/Evolucion-de-los-Smartphone>
- LYZANOR, Enrique (2012). Te explicamos todo sobre x86 y ARM: Diferencias, ventajas y para qué sirve cada uno. Recuperado el 21 de Abril de 2013, de <http://www.omicrono.com/2012/12/te-explicamos-todo-sobre-x86-y-arm-diferencias-ventajas-y-para-que-sirve-cada-uno/>
- MARTÍNEZ, María (2010), Telefonía Móvil. Recuperado el 29 de Junio de 2012, de <http://es.kioskea.net/contents/telephonie-mobile/reseaux-mobiles.php3>
- MERELO, Juan Julian (2004). Introducción al lenguaje XML. Recuperado el 1 de Agosto de 2012, de <http://geneura.ugr.es/~jmerelo/xml/>
- NICHOL, Scott (2004). Introduction to NuSOAP. Recuperado el 23 de Mayo de 2013, de <http://www.scottnichol.com/nusoapintro.htm>
- OCAÑA, Sandra (2009), Telefonía celular. Primera Generación. Recuperado el 27 de Junio de 2012, de <http://www.emagister.com/curso-telefonía-celular-movil-funcionamiento-generaciones/telefonía-celular-primera-generacion-1g>
- OCAÑA, Sandra. (2009). Telefonía celular (Móvil). Funcionamiento y Generaciones. Recuperado el 23 de Junio de 2012, de <http://dtm.unicauca.edu.co/pregrado/conmutacion/transp/5-TelefoníaMovil.pdf>
- OROZCO, Sergio - ORTIZ, Leticia (2012). Casos a incluir y casos a extender. Recuperado el 3 de Setiembre de 2013, de [http://www.milestone.com.mx/articulos/casos\\_a\\_incluir\\_casos\\_a\\_extender.htm](http://www.milestone.com.mx/articulos/casos_a_incluir_casos_a_extender.htm)
- POTENCIER, Fabien - ZANINOTTO, François (2012). Symfony 1.2, la guía definitiva. Recuperado el 14 de Agosto de 2012, de [http://www.librosweb.es/symfony\\_1\\_2/capitulo2/el\\_patron\\_mvc.html](http://www.librosweb.es/symfony_1_2/capitulo2/el_patron_mvc.html)
- QUIM, Liam (2012). Extensible Markup Language (XML). Recuperado el 1 de Agosto de 2012, de <http://www.w3.org/XML/>
- SAFFIRIO, Mario (2006). ¿Qué son los Web Services? Por qué los Web Services son hoy tan importantes. Recuperado el 11 de Agosto de 2012, de <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>
- THE PHP GROUP (2013). Documentation. Recuperado el 17 de Mayo de 2013, de <http://php.net/>

- TOBY (2009). ¿Para qué sirven los Web Services?. Recuperado el 12 de Agosto de 2012, de <http://turboprogramacion.blogspot.com.ar/2009/01/qu-sirven-los-web-services.html>
- TREB, Brandon (2008). iPhone Programming – Creating a ToDo List Using SQLite. Recuperado el 14 de Mayo de 2013, de <http://www.icodeblog.com/2008/08/19/iphone-programming-tutorial-creating-a-todo-list-using-sqlite-part-1/#sqlite3-framework>
- VERGE, S. (2011) iOS: A visual history. Recuperado el 26 de Junio de 2012, de <http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>
- W3C (2000). Simple Object Access Protocol (SOAP) 1.1. Recuperado el 11 de Agosto de 2012, de <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- WIKIPEDIA (2012). Servicio web. Recuperado el 11 de Agosto de 2012, de [http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web)
- WIKIPEDIA (2013). MySQL. Recuperado el 14 de Mayo de 2013, de <http://es.wikipedia.org/wiki/MySQL>
- WIKIPEDIA (2013). PHP. Recuperado el 17 de Mayo de 2013, de <http://es.wikipedia.org/wiki/PHP>
- YERKOCH (2009). ¿Web Service que es? ¿Para qué sirve?. Recuperado el 11 de Agosto de 2012, de <http://www.todoexpertos.com/categorias/tecnologia-e-internet/desarrollo-de-sitios-web/respuestas/2148128/web-service-que-es-para-que-sirve>
- APOLLO GROUP Inc.(2013) University Of Phoenix. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/us/app/university-of-phoenix-mobile/id429092408?mt=8>
- FCAPP (2011) Universidad De Jaen. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/es/app/universidad-de-jaen-app/id367619012?mt=8>
- HARVARD UNIVERSITY (2013)University Of Harvard. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/us/app/harvard-mobile/id389199460?mt=8>
- OMBIEL LIMITED (2013) University Of Bradford. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/gb/app/university-bradford-about/id391327000?mt=8>
- STANFORD UNIVERSITY (2013). University Of Stanford. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/us/app/istanford/id292922029?mt=8>
- THE UVIVERSITY OF MICHIGAN (2013) University Of Michigan. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/us/app/university-of-michigan/id380339596?mt=8>
- UNIVERSIDAD DE ALICANTE (2013) Universidad De Alicante. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/es/app/iua/id416776674?mt=8>
- UNIVERSIDAD DE NAVARRA (2012) Universidad De Navarra. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/nz/app/universidad-de-navarra/id483703213?mt=8>
- UNIVERSITY OF CALIFORNIA, SAN DIEGO, BUSINESS AFFAIRS (2013) University Of California, San Diego. Recuperado el 21 de Abril 2013, de

<http://mobile.ucsd.edu/>, <https://itunes.apple.com/app/uc-san-diego-ucsd/id318646412?mt=8>

- UNIVERSITY OF LIVERPOOL (2013). University Of Liverpool. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/gb/app/uolmobile/id528649971?mt=8>
- UNIVERSITY OF MIAMI (2012) University Of Miami. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/us/app/umiami/id445067887?mt=8>
- UNIVERSIDAD DE GRANADA (2012) Universidad De Granada. Recuperado el 21 de Abril 2013, de <https://itunes.apple.com/es/app/universidad-de-granada/id436735938?mt=8>

## **7. Anexos**

### **7.1. Anexo A: Apple Review Guidelines**

#### **1. Términos y condiciones**

- 1.1.** Como desarrollador de aplicaciones para la Tienda de Aplicaciones (o “AppStore”) usted deberá respetar los términos del Acuerdo de Licencia del Programa (ALP/PLA), de la Guía de Interfaz Humana, y de cualquier otro tipo de licencia o contrato entre usted y Apple. Las siguientes reglas y ejemplos procuran asistirlo para que su aplicación sea aceptada en la AppStore, no para modificar ni remover cláusulas de ningún otro acuerdo.

#### **2. Funcionalidad**

- 2.1.** Las aplicaciones que fallen serán rechazadas.
- 2.2.** Las aplicaciones que exhiban errores serán rechazadas.
- 2.3.** Las aplicaciones que no concuerden con lo publicitado por el desarrollador serán rechazadas.
- 2.4.** Las aplicaciones que incluyan ítems indocumentados u ocultos que sean inconsistentes con la descripción de la aplicación serán rechazadas.
- 2.5.** Las aplicaciones que hagan uso de APIs no públicas serán rechazadas.
- 2.6.** Las aplicaciones que lean o escriban datos fuera del área de contenedor que le ha sido asignada serán rechazadas.
- 2.7.** Las aplicaciones que descarguen códigos de manera alguna serán rechazadas.

- 2.8.** Las aplicaciones que instalen o presenten otro código ejecutable serán rechazadas.
- 2.9.** Las aplicaciones que representen versiones de “beta”, “demo”, “trial”, o “test” serán rechazadas.
- 2.10.** Las aplicaciones de iPhone también deben funcionar en iPad si modificación, bajo la resolución de iPhone, y a una resolución de 2X iPhone 3GS.
- 2.11.** Las aplicaciones que dupliquen otras aplicaciones ya existentes en la Tienda probablemente sean rechazadas, especialmente si hay muchas de ellas, tales como “pedos”, “eructos”, “linternas”, y aplicaciones de Kama Sutra.
- 2.12.** Aplicaciones que no sean muy útiles, que sean simplemente sitios web en forma de aplicaciones, o que no provean ningún tipo de entretenimiento constante podrían ser rechazadas.
- 2.13.** Las aplicaciones que sean primordialmente material de marketing o publicidad serán rechazadas.
- 2.14.** Las aplicaciones que tengan como objetivo presentar trucos o falsas funcionalidades que no sean claramente remarcadas como tales serán rechazadas.
- 2.15.** Las aplicaciones que sean mayores a 20MB de tamaño no podrán ser descargadas a través de redes de celular (esto está automáticamente prohibido por la Tienda de Aplicaciones).
- 2.16.** Las aplicaciones multifuncionales probablemente sólo puedan usar servicios en segundo plano para lograr sus propósitos: VoIP, audio playback, locación, finalización de tareas, notificaciones locales, etc.
- 2.17.** Las aplicaciones que requieran navegación en la web deberán usar el iOS WebKit Framework y WebKit JavaScript.
- 2.18.** Las aplicaciones que promuevan el consumo excesivo de alcohol u otras sustancias ilegales, o que inciten a los menores a beber alcohol o fumar, serán rechazadas.
- 2.19.** Las aplicaciones que provean un diagnóstico incorrecto u otros datos de dispositivo incorrectos serán rechazadas.

- 2.20. Los desarrolladores que se comporten como “spam” o abusen en la AppStore con versiones de aplicaciones similares serán removidos del Programa de Desarrolladores iOS.
- 2.21. Las aplicaciones que sean una simple canción o película deberían ser subidas a la tienda de iTunes. Las aplicaciones que sean simplemente un libro deberían ser subidas al iBookstore.
- 2.22. Las aplicaciones restrinjan arbitrariamente qué usuarios pueden usarlas, por ejemplo, por locación o carrera, probablemente sean rechazadas.
- 2.23. Las aplicaciones deben seguir los lineamientos de “iOS Data Storage” (almacenamiento de datos en iOS) o serán rechazadas (sistema de almacenamiento iCloud).
- 2.24. Las aplicaciones que se ofrezcan en el Quiosco (Newsstand) deben cumplir con los puntos del “Developer Program License Agreement” (Acuerdo de licencia del programa para desarrolladores). Éste hace referencia entre otras cosas que el desarrollador acuerda con Apple, ítems como usos permitidos y restricciones, obligaciones del desarrollador, restricciones en certificados, etc.
- 2.25. Las aplicaciones que muestran aplicaciones distintas a la suya para vender o promocionar de manera similar o para confundir en el AppStore, serán rechazadas.

### **3. Metadata**

- 3.1. Las aplicaciones o metadatos que mencionen el nombre de cualquier otra plataforma móvil serán rechazadas.
- 3.2. Las aplicaciones con texto marcador de posición (textos de relleno genéricos) serán rechazadas.
- 3.3. Las aplicaciones con descripciones que no sean relevantes para el contenido y la funcionalidad de la aplicación serán rechazadas.
- 3.4. Los nombres de las aplicaciones en iTunes Connect y en el dispositivo deben ser iguales o similares para evitar confusión.

- 3.5. Los iconos grandes y pequeños de la aplicación deben ser similares para evitar confusión.
- 3.6. Las aplicaciones con iconos y capturas de pantalla que no se corresponden a la clasificación por edades de 4+ será rechazada.
- 3.7. Las aplicaciones que tengan contenido no apropiado para las categorías y género seleccionadas serán rechazadas.
- 3.8. Los desarrolladores son responsables de asignar las calificaciones adecuadas para sus aplicaciones. La clasificaciones inapropiadas podrían ser cambiadas y/o eliminadas por Apple.
- 3.9. Los desarrolladores son responsables de asignar las palabras clave adecuadas para sus aplicaciones. Las palabras claves erróneas pueden ser cambiadas y/o eliminadas por Apple.
- 3.10. Los desarrolladores que intenten manipular o engañar a los comentarios de los usuarios o la tabla de clasificación en la AppStore con comentarios falsos o cualquier otros métodos inapropiados serán eliminados del Programa de Desarrolladores iOS.
- 3.11. Las aplicaciones que recomienden a los usuarios reiniciar sus dispositivos antes de su instalación o su ejecución podrían ser rechazadas.
- 3.12. Cuando se suban las aplicaciones para revisión, los enlaces URL tales como soporte, marketing y políticas de privacidad, deben estar en su entero funcionamiento.

#### **4. Localización**

- 4.1. Las aplicaciones que no notifiquen y obtengan permisos del usuario antes de recoger, transmitir y/o usar datos de localización serán rechazadas.
- 4.2. Las aplicaciones que utilizan APIs de localización para el control automático o autónomo de vehículos, aeronaves, u otros dispositivos serán rechazadas.
- 4.3. Las aplicaciones que utilizan las APIs de localización para la expedición, gestión de flotas, o servicios de emergencia serán rechazadas.

- 4.4.** Los datos de localización sólo se puede utilizar para proveer servicios directamente relacionados con las características que ofrece la aplicación para el usuario.

## **5. Notificaciones**

- 5.1.** Las aplicaciones que proporcionan notificaciones Push sin utilizar el Apple Push Notification (APN) de la API serán rechazadas.
- 5.2.** Las aplicaciones que utilizan el servicio de APN sin obtener una identificación de Apple para Aplicaciones Push, serán rechazadas.
- 5.3.** Las aplicaciones que envíen Notificaciones Push sin el consentimiento del usuario primero serán rechazadas.
- 5.4.** Las aplicaciones que envíen información personal o confidencial mediante notificaciones Push serán rechazadas.
- 5.5.** Aplicaciones que utilicen notificaciones Push para enviar mensajes no solicitados, o con el propósito de phishing o spam serán rechazadas.
- 5.6.** Las aplicaciones no pueden utilizar Notificaciones Push para enviar publicidad, promociones o marketing directo de ningún tipo.
- 5.7.** Las aplicaciones no pueden cobrar a los usuarios por el uso de notificaciones Push.
- 5.8.** Las aplicaciones que utilicen excesivamente la capacidad de la red o el ancho de banda del servicio APN o indebidamente sobrecarguen un dispositivo con notificaciones Push, serán rechazadas.
- 5.9.** Las aplicaciones que transmitan virus, ficheros, código de computadora o programas que puedan dañar o perturbar el funcionamiento normal del servicio APN, serán rechazadas.

## **6. Centro de Juegos**

- 6.1. Las aplicaciones que muestren cualquier ID de usuario al usuario final o de cualquier tercero será rechazada.
- 6.2. Las aplicaciones que usan IDs de los jugadores para cualquier uso que no sea aprobado por los términos de “Game Center” será rechazada.
- 6.3. Los desarrolladores que tratan de realizar búsquedas inversas, rastrear, relacionar, asociar o de cualquier otra forma explotar IDs de los jugadores, alias, u otra información obtenida a través del Game Center serán eliminados del Programa de Desarrolladores de iOS.
- 6.4. La información Game Center, como puntuaciones de clasificación, sólo se pueden utilizar en aplicaciones aprobadas para su uso con el Game Center.
- 6.5. Las aplicaciones que utilizan el servicio Game Center para enviar mensajes no solicitados, o con el propósito de phishing o spam serán rechazadas.
- 6.6. Las aplicaciones que utilizan excesivamente la capacidad de la red o ancho de banda del Game Center serán rechazadas.
- 6.7. Las aplicaciones que transmitan virus, ficheros, códigos de computadora o programas que puedan dañar o perturbar el funcionamiento normal del servicio Game Center serán rechazadas.

## **7. Publicidad**

- 7.1. Las aplicaciones que aumenten artificialmente el número de impresiones o de clics en los anuncios serán rechazadas.
- 7.2. Las aplicaciones que contengan banners iAd vacíos serán rechazadas.
- 7.3. Las aplicaciones que están diseñadas principalmente para la visualización de anuncios serán rechazadas.

## **8. Marcas e imagen comercial**

- 8.1. Las aplicaciones deben cumplir con todos los términos y condiciones explicados en la Guía para el uso de marcas comerciales y derechos de autor de Apple y la lista de marcas de Apple.
- 8.2. Las aplicaciones que indiquen que Apple es soporte o proveedor de las mismas, o que Apple respalda cualquier representación particular con respecto a la calidad o funcionalidad serán rechazadas.
- 8.3. Las aplicaciones que parecen confusamente similares a un producto de Apple existente o tema publicitario serán rechazadas.
- 8.4. Las aplicaciones que escriban mal los nombres de productos de Apple en su nombre de aplicación (por ejemplo: GPS para Iphone, iTunz) serán rechazados.
- 8.5. Las aplicaciones no pueden utilizar material protegido de terceros, como marcas comerciales, derechos de autor, patentes o términos de uso de terceros. La autorización para utilizar dicho material deberá proporcionarse mediante una petición.

## **9. Contenido multimedia**

- 9.1. Las aplicaciones que no utilizan el framework de MediaPlayer para acceder a los contenidos multimedia en la biblioteca de música serán rechazadas.
- 9.2. Las interfaces de usuario de las aplicaciones que imiten cualquier interfaz de iPod será rechazada.
- 9.3. El contenido de audio streaming que se use a través de una red celular no puede utilizar más de 5MB cada 5 minutos.
- 9.4. El contenido de video streaming que se use a través de una red celular más de 10 minutos debe utilizar HTTP Live Streaming e incluir una baseline de 64 kbps de sólo audio HTTP Live stream.

## **10. Interfaz de usuario**

- 10.1.** Aplicaciones deben cumplir con todos los términos y condiciones explicados en la “Apple iOS Human Interface Guideline” (Guía de interfaz humana de iOS), respecto al desarrollo de interfaces de usuario de la aplicación.
- 10.2.** Aplicaciones que se parecen a las aplicaciones cargadas en el iPhone, incluyendo la AppStore, iTunes Store, y iBookstore, se rechazarán.
- 10.3.** Las aplicaciones que no utilizan los elementos provistos por el sistema, como botones e iconos, correctamente y como se describe en las Directrices de Apple iOS interfaz humana serán rechazadas.
- 10.4.** Aplicaciones que crean o simulan entornos de la pantalla de inicio o experiencias de multi-aplicaciones o widgets serán rechazadas.
- 10.5.** Aplicaciones que alteran las funciones de los botones convencionales, tales como el volumen subir/bajar y el de llamada/silencio, serán rechazadas.
- 10.6.** Apple y nuestros clientes le dan gran valor a las interfaces simples, refinadas, creativas y bien pensadas. Ellas toman más trabajo, pero lo valen. Apple establece límite de calidad. Si la interfaz usuario es compleja o no supera dicho límite, podrían ser rechazadas.

## **11. Compras y moneda**

- 11.1.** Las aplicaciones que desbloqueen o activen funciones adicionales o funciones con otros mecanismos de la AppStore serán rechazadas.
- 11.2.** Las aplicaciones que utilicen un sistema distinto a la API de compra In-App Purchase (IAP) para comprar contenido, funcionalidades o servicios en una aplicación serán rechazadas.
- 11.3.** Las aplicaciones que utilicen IAP para comprar bienes físicos o bienes y servicios que se utilizan fuera de la aplicación serán rechazadas.
- 11.4.** Las aplicaciones que utilicen IAP para comprar créditos u otras monedas deben consumir esos créditos dentro de la aplicación.

- 11.5.** Las aplicaciones que utilicen IAP para comprar créditos u otras monedas que expiren serán rechazadas.
- 11.6.** Los contenidos con suscripciones utilizando IAP deben durar un mínimo de 7 días y estarán disponibles para el usuario desde todos sus dispositivos iOS.
- 11.7.** Las aplicaciones que utilicen IAP para comprar artículos deben asignar el tipo correcto de compra.
- 11.8.** Las aplicaciones que utilizan IAP para comprar el acceso a los componentes proporcionados por iOS incorporados en el dispositivo, tales como la cámara o el giroscopio, serán rechazadas.
- 11.9.** Las aplicaciones que contengan contenidos de tipo “alquiler” o servicios que expiren después de un tiempo limitado serán rechazadas.
- 11.10.** Los seguros de las aplicaciones deben ser libres, en cumplimiento legal en las regiones distribuidas, y no pueden usar IAP.
- 11.11.** En general, cuanto más cara su aplicación, más cuidadosamente la revisaremos.
- 11.12.** Las aplicaciones que ofrezcan suscripciones deben hacerlo a través de IAP, Apple compartirá el mismo ingreso dividido (70/30) con los desarrolladores para estas compras, como se establece en el Acuerdo de Licencia del Programa de desarrolladores.
- 11.13.** Las aplicaciones que enlacen con mecanismos externos para compras o suscripciones que se utilizarán en la aplicación, como por ejemplo un botón "Comprar" que vaya a un sitio web para comprar un libro digital, serán rechazadas.
- 11.14.** Las aplicaciones pueden leer o reproducir contenido aprobado (específicamente revistas, periódicos, libros, audio, música vídeo y almacenamiento de nube) que está suscrito o comprado fuera de la aplicación, siempre y cuando no haya un botón o enlace externo en la aplicación para comprar el contenido aprobado. Apple sólo recibirá una parte de los ingresos para el contenido adquirido dentro de la aplicación.
- 11.15.** Las aplicaciones sólo podrán utilizar renovación automática de las suscripciones a publicaciones periódicas (periódicos, revistas), aplicaciones de negocio (empresa,

productividad, profesional creativo, almacenamiento en la nube) y aplicaciones multimedia (video, audio, voz), o serán rechazadas.

- 11.16.** Las aplicaciones pueden habilitar funciones adicionales aprobadas o funcionalidades cuando se usan en conjunto con productos físicos específicos aprobados (como un juguete), siempre y cuando las características y funciones adicionales sean completamente dependientes del hardware (por ejemplo, una aplicación que se utiliza para controlar un telescopio). También pueden estar disponibles a través de la aplicación mediante el uso de IAP.

## **12. Scraping and aggregation**

- 12.1.** Las aplicaciones que utilizan scraping (raspado) de cualquier información de los sitios de Apple (por ejemplo, de apple.com, iTunes Store, AppStore, iTunes Connect, los programas de desarrollo de Apple, etc) o creen rankings utilizando el contenido de los sitios y servicios de Apple serán rechazadas.
- 12.2.** Las aplicaciones pueden utilizar canales RSS aprobados por Apple, tal como el iTunes Store Feed RSS.
- 12.3.** Las aplicaciones que son simplemente web clippings (recortes de internet), agregadores de contenido, o una colección de enlaces, pueden ser rechazadas.

## **13. Daños en el dispositivo**

- 13.1.** Las aplicaciones que estimulen a los usuarios a utilizar un dispositivo de Apple en una forma que pueda causar daños en el mismo serán rechazadas.
- 13.2.** Las aplicaciones que consuman rápidamente la batería del dispositivo o generen calor excesivo serán rechazadas.

## **14. Ataques personales**

**14.1.** Cualquier aplicación que sea difamatoria, ofensiva, mal intencionado, o que puedan poner a un individuo o grupo en peligro será rechazada.

**14.2.** Profesionales satíricos políticos y humoristas están exentos de la prohibición de comentario ofensivo o mal intencionados.

## **15. Violencia**

**15.1.** Las aplicaciones que retraten imágenes realistas de personas o animales muertos o mutilados, fusilados, apuñalados, torturados o heridos serán rechazadas.

**15.2.** Las aplicaciones que muestran la violencia o el abuso de niños serán rechazadas.

**15.3.** "Enemigos" en el contexto de un juego no pueden apuntar a una determinada raza, cultura, gobierno o corporación verdadera, o cualquier otra entidad real.

**15.4.** Aplicaciones que implican representaciones realistas de armas, de tal forma que se fomente el uso ilegal o irresponsable de armas serán rechazadas.

**15.5.** Aplicaciones que incluyen juegos de Ruleta Rusa serán rechazadas.

## **16. Contenido objetable**

**16.1.** Las aplicaciones que presenten contenido objetable o excesivamente crudo serán rechazadas.

**16.2.** Las aplicaciones que estén diseñadas principalmente para molestar o disgustar a los usuarios serán rechazadas.

## **17. Privacidad**

**17.1.** Las aplicaciones no pueden transmitir datos de un usuario sin obtener el permiso previo del usuario y proporcionar al usuario acceso a la información sobre cómo y dónde los datos se utilizarán.

**17.2.** Las aplicaciones que requieran que los usuarios compartan información personal, como su dirección de correo electrónico y fecha de nacimiento, con el fin de funcionar serán rechazadas.

**17.3.** Las aplicaciones para que estén destinadas para recolección de datos de menores de edad serán rechazadas.

## **18. Pornografía**

**18.1.** Las aplicaciones que contienen material pornográfico, definida por el Diccionario Webster como "descripciones explícitas o que muestren los órganos sexuales o actividades destinadas a estimular erotismo en lugar de sentimientos emocionales", serán rechazadas.

**18.2.** Las aplicaciones que contengan contenido generado por usuarios que creen con frecuencia contenido pornográfico (aplicaciones tipo "Chat Roulette") serán rechazadas.

## **19. Religión, cultura y etnia**

**19.1.** Las aplicaciones que contienen referencias o comentarios acerca de un grupo religioso, cultural o étnico que sean difamatorios, ofensivos, mal intencionados o que pueda exponer al grupo objetivo a violencia o daño serán rechazadas.

**19.2.** Las aplicaciones pueden contener o citar textos religiosos siempre y cuando las traducciones sean exactas y no engañosas. Los comentarios deben ser educativos o informativos y no difamatorios.

## **20. Concursos, sorteos, loterías y rifas**

**20.1.** Los sorteos y los concursos deben ser patrocinados por el desarrollador o compañía desarrolladora de la aplicación.

**20.2.** Las reglas oficiales de sorteos y concursos, se deben presentar en la aplicación y dejar en claro que Apple no es patrocinador o participante en la actividad de cualquier manera.

- 20.3.** Debe estar permitido por ley que el desarrollador pueda ejecutar una aplicación de lotería. Una aplicación de lotería debe contar con las siguientes características: consideraciones, azar, y un premio.

## **21. Caridad y contribución**

- 21.1.** Las aplicaciones que incluyan la posibilidad de hacer donaciones a organizaciones caritativas reconocidas deben ser sin costo.
- 21.2.** La recolección de las donaciones deben hacerse a través de un sitio web en Safari o un SMS.

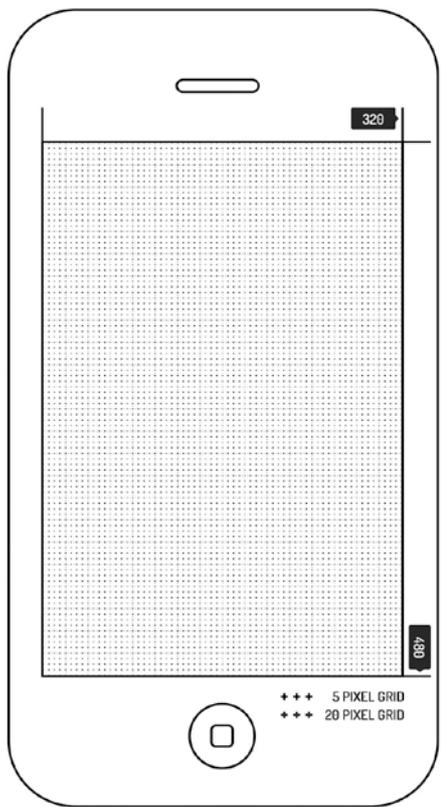
## **22. Requisitos legales**

- 22.1.** Las aplicaciones deben cumplir con todos los requisitos legales en todos los lugares donde se ponen a disposición de los usuarios. Es obligación del desarrollador entender y cumplir con todas las leyes locales.
- 22.2.** Las aplicaciones que posean representaciones falsas, fraudulentas o engañosas serán rechazadas.
- 22.3.** Las aplicaciones que soliciten, promuevan o fomenten un comportamiento criminal o claramente imprudente serán rechazadas.
- 22.4.** Las aplicaciones que permiten el intercambio de archivos ilegales serán rechazadas.
- 22.5.** Las aplicaciones que están diseñados para ser utilizados como ayudas ilegales de juego, incluidos contadores de cartas, serán rechazadas.
- 22.6.** Las aplicaciones que permitan llamadas telefónicas anónimas o travesura por mensajes SMS / MMS serán rechazadas.

- 22.7.** Los desarrolladores que creen aplicaciones que tratan de descubrir las contraseñas de usuario u otros datos privados, explícitamente o no, se eliminarán del Programa de Desarrolladores de iOS.
- 22.8.** Las aplicaciones que contengan puntos de control de DUI que no estén publicados por las agencias de aplicación de leyes, o alienten o permitan conducir en estado de ebriedad, se rechazarán.

Este documento representa nuestros mejores esfuerzos para compartir cómo se revisan las aplicaciones presentadas a la AppStore, y esperamos que sea una guía útil al desarrollar y presentar aplicaciones. Se trata de un documento vivo que evolucionará a medida que se presenten nuevas aplicaciones y situaciones. Lo actualizaremos periódicamente para reflejar estos cambios.

## 7.2. Anexo B: Mockups



▲ PROYECTO: \_\_\_\_\_

📄 PANTALLA: \_\_\_\_\_

👤 AUTOR: \_\_\_\_\_

✍️ NOTAS: \_\_\_\_\_

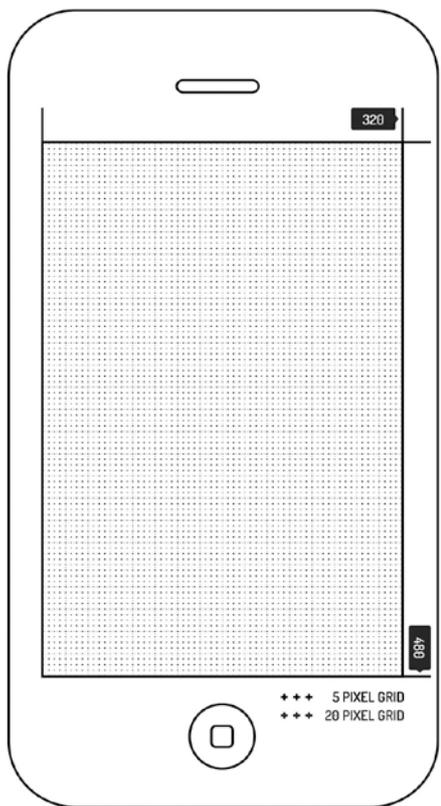
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



▲ PROYECTO: \_\_\_\_\_

📄 PANTALLA: \_\_\_\_\_

👤 AUTOR: \_\_\_\_\_

✍️ NOTAS: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_