



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CONTADURIA Y ADMINISTRACION

**DESARROLLO DE UNA APLICACION PARA DISPOSITIVO
MOVIL PARA GENERAR REPORTES DE SOPORTE TECNICO
EN INFORMATICA DENTRO DE UNA ORGANIZACION.**

**DISEÑO DE UN SISTEMA O PROYECTO
PARA UNA ORGANIZACION**

FERNANDO ANDRES MARTINEZ FABIAN



MÉXICO, D.F.

2013



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CONTADURIA Y ADMINISTRACION

**DESARROLLO DE UNA APLICACION PARA DISPOSITIVO
MOVIL PARA GENERAR REPORTES DE SOPORTE TECNICO
EN INFORMATICA DENTRO DE UNA ORGANIZACION.**

**DISEÑO DE UN SISTEMA O PROYECTO
PARA UNA ORGANIZACION**

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMATICA

PRESENTA:

FERNANDO ANDRES MARTINEZ FABIAN

ASESOR:

DRA. GRACIELA BRIBIESCA CORREA



MÉXICO, D.F.

2013

AGRADECIMIENTOS

Este Proyecto ha sido el resultado de bastante tiempo de estudio y paciencia de aprender un tema que ha resultado ser bastante innovador en la vida actual de las personas, que son los Dispositivos Móviles, así como su entorno de Desarrollo de Programación de las Aplicaciones.

Quiero agradecer a todos los que han confiado en mi, así como también a quienes han dejado tanto de si mismos en este Proyecto, y voy a hacerlo cronológicamente al igual que comenzó este viaje, en mi hogar:

A ti Mamá, gracias por esas largas horas de platica conmigo, por tus consejos, observaciones, así como de las llamadas de atención cuando me las merecía, por apoyarme en las decisiones importantes e ir hacia adelante, por tu entusiasmo y por tu lucha, has sido esencial en este Proyecto.

A ti Papá, gracias por tu apoyo, así como también por cubrir todos los gastos para que estudiara desde que era un niño hasta concluir la Licenciatura en Informática.

A mi Tía Ma. del Carmen Fabián Orduña y a mi Tío José Manuel Fabián Orduña, gracias por todo su apoyo en los buenos momentos, así como también en los malos momentos.

A mis amigos de la Facultad, gracias por esos buenos momentos en los cuales nos divertimos bastante, así como cuando se trataba de estudiar juntos, siempre supimos salir delante de cualquier dificultad que se nos presentaba, nunca los olvidaré.

A mis Maestros de la Facultad, gracias por el aporte de sus conocimientos e ideas, los cuales aproveché bastante durante toda la Licenciatura, para tener unas bases sólidas para el futuro.

A mi Asesora, la Dra. Graciela Bribiesca Correa, gracias por su valioso tiempo para la búsqueda y selección del tema de la Tesis. Muchas Gracias por la ilusión, el optimismo y sus consejos, los cuales hicieron posible que este Proyecto se haya concretado con éxito.

A la Facultad de Contaduría y Administración, gracias por darme la oportunidad de haber estudiado en esta Escuela, me llevo guardado en mi corazón muchos momentos en los cuales conocí a muchas personas con las cuales siempre estaré en contacto con ellas, así como también aprendí bastante de mis Maestros.

A la Universidad Nacional Autónoma de México, *mi Alma Mater*, a la cual siempre tendré un cariño y respeto por siempre, ya que tuve la oportunidad de estudiar en la mejor Universidad de mi País.

Y siempre, gracias a **Dios** por ser el eje de mi vida, por darme la posibilidad de estudiar y de sentir su amor en cada minuto de mi existencia.

Fernando Andrés Martínez Fabián

INDICE

AGRADECIMIENTOS	3
INDICE	5
INTRODUCCION.....	7
CAPITULO No. 1: “<u>MARCO DE REFERENCIA</u>”	17
1.1 Dispositivos Móviles:	18
1.2 Programación Web:	20
1.3 Programación de Aplicaciones de Escritorio:	21
1.4 Sistemas Distribuidos:	22
CAPITULO No. 2: “<u>HISTORIA DE LOS DISPOSITIVOS MOVILES</u>”	26
2.1 Smartphone:	31
2.2 PDA:	31
2.3 Dispositivos Móviles en el Trabajo:	32
2.4. Sistemas Operativos para Dispositivos Móviles:	32
CAPITULO No. 3: “<u>APPLE, INC.</u>”	39
3.1 Historia de Apple, Inc.:	40
3.2. Historia del Mac OS:	45
CAPITULO No. 4: “<u>PLATAFORMA DE DESARROLLO DEL IPHONE Y EL IPAD</u>”	53
4.1 Xcode 4:	54
4.2 Desarrollo de Aplicaciones para el iOS:	61
4.3 iOS Cocoa Touch:	62
4.4 Gestión de Datos (iOS):	63
4.5 Desarrollo para OS X (El Sistema Operativo más avanzado del mundo):	66
4.6. Mac OS X Cocoa:	68
4.7 Proceso de Desarrollo de una Aplicación para el iPhone y el iPad:	70
4.8 Historia :	76
CAPITULO No. 5: “<u>ANALISIS Y DISEÑO DEL SISTEMA</u>”	77
5.1 Análisis del Sistema:	78
5.2 Diagramas de Flujo:	84
5.3 Diseño de la Base de Datos:	91
5.4 Diagramas de Clases:	92

CAPITULO No. 6: “<u>DESARROLLO DE LAS APLICACIONES</u>: iShelperApp Y iShelperForMac“	95
6.1 10 Sencillos Pasos para Desarrollar una Aplicación para Dispositivos Móviles: ...	96
6.2 CRUD (ABM):	98
6.3 Desarrollo de la Aplicación “iShelperApp” para el iPad:.....	99
6.4 Desarrollo de la Aplicación “iShelperForMac” para iMac y Macbook:	131
CAPITULO No. 7: “IMPLEMENTACION DE LAS APLICACIONES: iShelperApp Y iShelperForMac“	143
7.1. iShelperApp:	145
7.2 iShelperForMac:	164
CONCLUSIONES Y RECOMENDACIONES	169
8.1 Conclusiones:	170
8.2 Recomendaciones:	171
ANEXO A: “DISPOSITIVOS DE APPLE, INC.”	172
ANEXO B: “PROGRAMAS DE APPLE, INC.”	183
ANEXO C: “<u>HELP DESK</u>”	194
ANEXO D: “<u>WI-FI</u>”	201
GLOSARIO DE TERMINOS TECNICOS	209
BIBLIOGRAFIA	217

INTRODUCCION

La comunicación se ha convertido, con el paso del tiempo, en una gran necesidad del ser humano para expresarse de forma rápida y con mayor calidad. Se define como el medio para transmitir información que va de un emisor hacia un receptor, con todo esto ha surgido la invención de un sinnúmero de aparatos que son capaces de realizar dicha función.

De esta manera, la evolución de los medios de comunicación ha pasado desde la escritura hasta medios que permiten el envío de información a distancia como el telégrafo y el teléfono convencional. Para finales del siglo XIX, surgió un concepto nuevo: la comunicación inalámbrica. Guglielmo Marconi fue uno de los precursores de esta nueva tecnología, mediante su invento que es el radio. Posteriormente, en el año de 1921, surgió el primer servicio de telefonía móvil (SMT por sus siglas en inglés), que fue utilizado por el departamento de la policía en la Ciudad de Detroit.

En la última década, la telefonía celular ha tenido un gran auge. Ya que no solamente sirve como medio de transmisión de voz de una persona a otra, sino que ahora se están utilizando cada vez más otras funciones que ofrece un teléfono celular. Así que ahora, es posible tener a nuestro alcance otro tipo de información como es el texto, las imágenes, audio y video.

Considerando que vivimos en un mundo que vive de prisa, la optimización del tiempo es sumamente importante. Si se logra reducir el tiempo que se emplea para la realización de diversas actividades, ya sea llevándolas a cabo de forma simultánea o eliminando algunos factores que representen una pérdida de tiempo, como es el caso de la distancia, es posible efectuar un mayor número de actividades.

La Importancia de la Información:

La Información es el conjunto de datos que tiene un significado y relevancia para alguna entidad, ya que ésta tiene un valor, el cual, dependiendo del contexto en el que se encuentre, puede llegar a ser muy grande.

Existen muchas situaciones históricas en las que la posesión de cierta información ha sido crucial para el éxito de distintas empresas, como son negocios, desarrollos tecnológicos, etc., ya que el hecho de poseer la información de manera confiable y en el momento preciso permitió tomar la mejor decisión.

Para que la información pueda ser útil tiene que clasificarse y ordenarse de alguna manera, es decir, debe existir un Sistema de Información, el cual es un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución, también pueden ayudar a los administradores y al personal para analizar problemas, visualizar cuestiones complejas y desarrollar nuevos productos.

El Manejo de la Información:

El manejo de la información requiere el desarrollo de determinadas capacidades en cada persona, a continuación se mencionan las más importantes para llevar con éxito este proceso, ya que en cada una de ellas se realizan acciones y procesos, que a su vez, perfilan otras habilidades:

1. Determinar las necesidades de la información:

- Partir de algún interés o necesidad.
- Cuestionar la información.
- Seleccionar un punto de vista.
- Definir de manera correcta lo que se quiere saber.
- Definir a qué tipo de público va a ir enfocado la información.

2. Planear la búsqueda de información:

- Definir los objetivos de acuerdo a las necesidades de la información.
- Realizar un plan de actividades.
- Reconocer las instancias que manejan la información.
- Tener en cuenta la extensión del tema.
- Contar con un método de sistematización de la información como es el registro, la concentración, la clasificación, la organización y la jerarquización.
- Definir estrategias y métodos de búsqueda.

3. Usar estrategias de búsqueda:

- Identificar palabras clave, temas y subtemas.
- Conocer y aplicar técnicas de lectura rápida.
- Emplear el subrayado como un recurso en la lectura.
- Usar buscadores, así como los índices temáticos.
- Dirigirse a personas que conocen acerca del tema.

4. Identificar y registrar las fuentes:

- Buscar lugares en donde esté disponible la información y revisar su nivel de accesibilidad.
- Saber lo que se puede encontrar de información en cada lugar.
- Saber que es una fuente.
- Evaluar la confiabilidad de las fuentes.
- Distinguir la fuente de información de cada medio.
- Hacer fichas que contengan datos bibliográficos (referencias), ideas principales (registros de información) y la utilidad (aplicación y uso de la información).

5. Evaluar la información obtenida:

- Emplear la síntesis de información.
- Distinguir lo general y lo particular de la información.
- Usar criterios para captar, seleccionar, integrar y organizar.
- Validar la información con racionalidad.
- Distinguir hechos de opiniones.
- Revisar lo que se ha hecho.
- Ser capaz de ver la forma en que va evolucionando el proyecto.

6. Procesar para producir la información propia:

- Seleccionar las herramientas adecuadas para procesar la información.
- Dominar y aplicar el análisis, para realizar la síntesis de la información.
- Ser capaz de reflexionar y concluir.
- Aplicar los elementos básicos de expresión escrita.
- Utilizar esquemas, cuadros sinópticos y/o tablas.

7. Generar productos de comunicación de calidad:

- Generar productos comunicativos que responden a las necesidades reales de información o comunicación.
- Diseñar los productos comunicativos para responder con seguridad a las necesidades.
- Prever en el diseño los formatos, estructuras u organización de los contenidos que hacen transparente la información.
- Emplear la información en la forma de los medios que más le convenga a los usuarios.

8. Evaluar el proceso y los productos:

- Ser objetivo consigo mismo.
- Identificar la utilidad del producto.
- Identificar lo aprendido y reconocer su utilidad para seguir aprendiendo.
- Reconocer su propia eficiencia.
- Verificar que el producto sea comprensible para el destinatario.

¿Cómo resolver el Manejo de la Información los Dispositivos Móviles?

También resulta necesario controlar algunas situaciones estando en diferentes lugares, es decir, tener el control de la información a través de la distancia. En conclusión, el tiempo se ha convertido en un elemento valioso y por lo tanto, el hecho de encontrar la forma de recuperarlo u optimizarlo es lo que impulsa al desarrollo de nuevas herramientas e inventos.

De esta manera, surgen dispositivos que permiten al hombre tener el control de la vida cotidiana y que a la vez le facilitan la realización de sus actividades. Estos nuevos aparatos han cubierto diferentes tipos de necesidades. Conforme transcurre el tiempo, cada vez se van reduciendo de tamaño y al mismo tiempo incrementan sus capacidades de procesamiento y memoria. Entonces, los dispositivos móviles ya no están limitados a realizar una sola función. Por ejemplo, un teléfono celular, antes su función era la de un medio de comunicación portable; ahora contiene diferentes elementos como agenda, juegos, cámara fotográfica, y son capaces de recibir datos multimedia, video y audio; así como también pueden servir para monitorear lugares.

Es necesario contar con la infraestructura adecuada que permita la transmisión de datos en los dispositivos móviles, pero ahora también ofrecen la capacidad de desarrollar aplicaciones personalizadas para cada usuario. De tal forma que es indispensable contar con una herramienta que permita el desarrollo de dichas aplicaciones.

Ventajas de los Dispositivos Móviles:

A continuación se mencionan las Ventajas de trabajar con los dispositivos móviles:

- Por su tamaño, son fáciles de transportar.
- Su costo es menor.
- Realizan varias funciones.
- Al poder llevarlos a varias partes, ayudan en la organización personal.
- Cuentan con batería de larga duración.
- Algunos de ellos cuentan con rastreo satelital, brindando inclusive seguridad al usuario.
- Tienen conexión a Internet vía inalámbrica.
- Son de fácil acceso para cuando se tenga que hacer uso de ellos.

La Importancia del Trabajo de Titulación, el cual se refiere al Desarrollo de Aplicaciones para Dispositivos Móviles:

El desarrollo de aplicaciones para dispositivos móviles, no debe variar de los pasos a seguir cuando se elaboran aplicaciones para las computadoras de escritorio, lo único que va a cambiar es la última parte, la cual es la compilación o generación de código ejecutable sobre el sistema operativo contenido en el dispositivo.

A continuación se establecen los siguientes pasos:

a) Análisis de Requerimientos:

El programador deberá determinar, las necesidades que los usuarios finales tienen y los requerimientos que se les pedirá tener a la aplicación.

b) *Diseño de la Aplicación:*

Es importante tener en cuenta que para este tipo de aplicaciones, el de crear programas separados por cada uno de los posibles usos que se le de a la aplicación.

En cuanto al diseño de la interfaz de usuario, se tiene que decidir la correspondencia entre la aplicación y la pantalla.

Finalmente, se debe tener en cuenta que dentro del diseño los aspectos que están relacionados con la conectividad y con la entrada/salida, ya que son puntos muy importantes que van a determinar la portabilidad de la aplicación.

c) *Desarrollo de la Aplicación:*

Es en esta parte en la que el desarrollador se dedica a programar la aplicación para que cumpla con los requerimientos especificados; ya que se dedicará a escribir el código, diseñar las pantallas, adaptar los controles y se harán las pruebas hasta que la aplicación esté lista.

Existen solo una forma de desarrollar las aplicaciones para los dispositivos móviles: que es desde una computadora de escritorio o laptop (ya sea una PC o una computadora Mac).

Los lenguajes de programación y sus entornos de desarrollo asociados disponibles para el desarrollo de aplicaciones para los dispositivos móviles son los mismos que se utilizan para las demás computadoras, no obstante existen algunos lenguajes que están diseñados para ello.

Siempre existe una solución de compromiso entre requerimientos de cómputo, facilidad de uso y de aprendizaje, costo y calidad de los resultados que se obtienen. Al desarrollar las aplicaciones para los dispositivos móviles hay que tener en cuenta su especificidad. **Los dispositivos móviles no son una computadora de escritorio, ni una laptop.**

Versatilidad:

La Versatilidad que brindan los dispositivos móviles, que cada día surgen más aplicaciones, son una herramienta vital que sirven para optimizar los tiempos de trabajo, ya que permiten la realización de distintas actividades de oficina, o la posibilidad de establecer comunicaciones mientras se encuentren de viaje o resolviendo otras cuestiones.

Un uso adecuado de estos recursos tecnológicos, combinado con la comunicación de persona a persona, siguen siendo imprescindibles y que nunca se podrán dejar de lado, ya que con esto surge el nacimiento de un nuevo paradigma en el campo de la comunicación empresarial.

Se han desarrollado en los últimos años, diversas aplicaciones potentes para los dispositivos móviles que son específicas para los sectores profesionales de todo tipo, como pueden ser: logística, ventas, sector salud, servicios técnicos, etc.

Comunicación entre los Dispositivos Móviles con las Computadoras de Escritorio o Laptops:

A continuación se explica brevemente en qué consiste cada uno de los tipos de comunicación entre los dispositivos móviles y las computadoras de escritorio:

- Infrarrojos:

Permiten la comunicación entre dos nodos, usando una serie de leds infrarrojos para ello. Se trata de emisores y receptores de las ondas infrarrojas entre ambos dispositivos; se utiliza principalmente para realizar el intercambio de datos entre los dispositivos móviles, como PDAs o teléfonos celulares, ya que el rango de velocidad y el tamaño de los datos a enviar y recibir es pequeño.

- Bluetooth:

Es un estándar global de comunicación inalámbrica de corto alcance, para transmitir voz y datos entre los dispositivos móviles como son los teléfonos celulares, laptops y computadoras de escritorio, mediante un enlace de radiofrecuencias. Su capacidad de transferencia es de 720 kbps, y tiene un alcance entre 10 y 100 metros; es multidireccional y puede atravesar paredes y otras barreras que no son metálicas.

- Hi-Fi:

Es una norma de calidad que explica lo siguiente: la reproducción del sonido o imágenes es bastante fiel al original, la alta fidelidad pretende que los ruidos y la distorsión sean mínimos. El Hi-Fi tiene como propósito brindar redes inalámbricas de alta calidad para el manejo y el control de archivos (música y videos), además de que se utiliza para las videoconferencias sin interrupción.

- Wi-Fi:

Permite la creación de ondas de radio en frecuencias sin licencia, lo que evita al operar el problema de las licencias o la autorización del agente regulador de las comunicaciones. Básicamente, se necesitan dos cosas en una conexión inalámbrica: un "hotspot" y un dispositivo con capacidad de comunicación Wi-Fi. Hotspot es el punto de acceso por el cual se transmite la señal. Actualmente, muchas computadoras vienen con un sistema de comunicación Wi-Fi integrado.

Además, en varios hogares, oficinas, centros comerciales, escuelas, restaurantes, librerías, hospitales, etc., poseen sus propias redes Wi-Fi, para que la conexión a Internet, sea sin cables. Estas redes, mejor conocidas como WLANs, permiten el acceso a Internet de Alta Velocidad en distancias menores de 100 metros, o sea, relativamente pequeñas.

- **Wi-Max:**

Es una de las tecnologías con mayor Ancho de Banda para la transmisión y recepción de datos, es una Norma de transmisión por ondas de radio de última generación que permite la recepción de datos por microondas y retransmisión por ondas de radio. Está definido bajo la Norma 802.16 MAN, que es un protocolo para Redes de Área Metropolitana, el cual proporciona acceso conjunto con varios repetidores de señal superpuestos para que de esta forma, ofrecer una cobertura total con promedio de un radio de 50 km, y con velocidades de hasta 124 Mbps.

Objetivo de este Trabajo de Titulación:

Desarrollar un sistema de información para el control de soporte técnico dentro de una organización, mediante el uso de un dispositivo móvil.

Alcance de este Trabajo de Titulación:

El Alcance de la Aplicación para Dispositivo Móvil (iPad) para llevar el control de soporte técnico, se desarrolló para elaborar los reportes de soporte técnico, así como también la base de datos para tener en orden dichos reportes en el Área de Soporte Técnico de la Empresa.

Resumen Capítular:

Introducción:

Dentro de este capítulo, se explica brevemente la importancia de la información, así como el manejo de la misma, también se explica la utilización de los dispositivos móviles para resolver el problema del manejo de la información y las ventajas que tienen los dispositivos móviles.

Capítulo No. 1 “Marco de Referencia”:

Dentro de este segundo capítulo, se explican las características, las ventajas y desventajas de los dispositivos móviles, la programación web, las aplicaciones de escritorio y los sistemas distribuidos, también el modelo cliente-servidor y la utilización de las aplicaciones distribuidas en los dispositivos móviles.

Capítulo No. 2 “Historia de los Dispositivos Móviles”:

Dentro de este tercer capítulo, se explica la historia de los dispositivos móviles, así como las definiciones de Smartphone y PDA, también la utilización de los dispositivos móviles en el trabajo y los Sistemas Operativos que tienen los dispositivos móviles.

Capítulo No. 3 “Apple, Inc.”:

Dentro de este cuarto capítulo, se explica la historia de la Empresa “Apple, Inc.” y del Sistema Operativo Mac OS.

Capítulo No. 4 “Plataforma de Desarrollo del iPhone y el iPad”:

Dentro de este quinto capítulo, se explica a detalle la plataforma de desarrollo de aplicaciones para los Dispositivos Móviles y las Computadoras de Apple, Inc. como es el programa Xcode 4, también se explica en que consiste el desarrollo de aplicaciones para iOS y Mac OS X, así como también el proceso de desarrollo de una aplicación para el iPhone, iPod Touch y el iPad.

Capítulo No. 5 “Análisis y Diseño del Sistema”:

Dentro de este sexto capítulo, se explica con detalle el Análisis del Sistema el cual consiste en la definición de soporte técnico, su vínculo con las tecnologías de la información y las comunicaciones, así como el mantenimiento del equipo de cómputo, los cuales consisten en el mantenimiento preventivo, correctivo y predictivo, también se explican los diagramas de flujo del mantenimiento preventivo y correctivo, el diseño de la base de datos y los diagramas de clases.

Capítulo No. 6 “Desarrollo de las Aplicaciones: iShelperApp y iShelperForMac”:

Dentro de este séptimo capítulo, se explican los 10 pasos para desarrollar una aplicación para dispositivos móviles, también la definición de CRUD –Create, Read, Update, Delete- (ABM –Alta, Baja, Modificación- en español), así como la explicación a detalle del desarrollo de las aplicaciones: “iShelper_App” para el iPad y “iShelper_For_Mac” para iMac y MacBook.

Capítulo No. 7 “Implementación de las Aplicaciones: iShelperApp y iShelperForMac”:

Dentro de este octavo capítulo, se explica la implementación de las aplicaciones: “iShelperApp” para el iPad y “iShelperForMac” para iMac y MacBook.

Conclusiones y Recomendaciones:

Dentro de este último capítulo, se explican las Conclusiones, así como también las Recomendaciones del trabajo de titulación para realizarse en un futuro cercano.

CAPITULO No. 1: “MARCO DE REFERENCIA”

1.1 Dispositivos Móviles:

¿Qué es un Dispositivo Móvil?

Son aparatos de tamaño pequeño que cuentan con algunas capacidades de procesamiento, también cuentan con conexión ya sea permanente o intermitente a una red, cuentan con memoria limitada, están diseñados específicamente para una sola función, pero pueden llevar a cabo otras más.

Características:

Las principales características son las siguientes:

- Portabilidad.
- Son de tamaño pequeño.
- Tienen una limitada capacidad de procesamiento.
- Pueden tener o no una conexión a una red.
- Su memoria es limitada.
- Está diseñados para una función en específico, pero se le pueden agregar otras más.
- Son utilizadas de forma individual.

Otra característica es que los dispositivos son inalámbricos, esto es que son capaces de comunicarse o de acceder a una red, sin la necesidad de conectarse con cables, por ejemplo un PDA o un Smartphone. Este tipo de dispositivos parecieran que estuvieran conectados a una red mediante un cable, pero su tecnología es inalámbrica.

Algunas de las características que hacen de estos dispositivos ser diferentes con respecto a las computadoras de escritorio, son las siguientes:

- Tienen una funcionalidad limitada.
- No es necesario que sean extensos, así como también que estén actualizados.
- En poco tiempo, el usuario deberá de cambiarlo por otro aparato que sea más moderno.
- Son más accesibles en cuanto a su Costo.
- Son fáciles de utilizar.
- No se requiere ser un usuario experto.

Los dispositivos móviles que entran en la Categoría son:

- Asistentes Personales Digitales (PDA, iPod Touch, etc.)
- Teléfonos celulares Inteligentes (iPhone, BlackBerry, etc.) que su Sistema Operativo es similar al de las PDA.
- Teléfonos celulares con capacidad de ejecución de programas del entorno Java.
- Tablets

Clasificación:

Por su definición y características de los dispositivos móviles, se pueden clasificar en Cuatro Grupos, que son los siguientes:



Figura 1.1: Clasificación de los Dispositivos Móviles

¿Para qué sirve un Dispositivo Móvil?

La expansión de las computadoras personales a nuestro alrededor hacen que se vuelvan una herramienta de trabajo necesaria, y también de diversión. Pero conforme pasa el tiempo, las computadoras se vuelven cada vez más poderosas, y brindan a los usuarios varias posibilidades, lo que hace que no sólo estén presentes en el trabajo y el hogar, sino que también se presenten en cualquier ámbito de nuestra vida.

Por lo tanto, somos testigos de que con el paso del tiempo, los fabricantes se han centrado en fabricar aparatos cada vez más pequeños, otorgándoles la habilidad de comunicarse y también de hacer cálculos. Sin importar qué tipo de aparato sea, necesitamos que tengan aplicaciones similares para poder utilizarlas en cualquier momento que lo queramos.

Ventajas y Desventajas de los Dispositivos Móviles:

Ventajas:

- Al ser de tamaño pequeño, son fáciles de transportar.
- El costo es menor.
- Actualmente los dispositivos móviles realizan diversas funciones.
- Al poder llevarlos a cualquier parte, ayudan en la organización personal.
- Cuentan con batería, así que no es necesario conectarlos a la corriente eléctrica.
- Algunos de ellos, cuentan con rastreo satelital, brindando inclusive seguridad al usuario.
- Se pueden conectar a Internet en cualquier lugar.
- Son de fácil acceso para cuando uno los necesite utilizar.

Desventajas:

- Cuentan con memoria limitada.
- La capacidad de procesamiento es menor.
- En poco tiempo, habrá que cambiar de aparato por uno más moderno.
- La batería tiene un tiempo de energía limitado.
- Algunas pantallas no tienen la suficiente resolución para navegar en Internet.
- Algunos no tienen la misma capacidad para que sean utilizados en el trabajo.

1.2 Programación Web:**Internet:**

Es un método de interconexión de redes de computadoras que se encuentra implementado en un conjunto de protocolos denominado TCP/IP, y garantiza que las redes físicas heterogéneas funcionen como una red (lógica) única. Por esa razón, Internet también se le conoce como “red de redes”, es importante destacar que Internet no es un nuevo tipo de red física, sino que es un método de interconexión. Surgió en el año de 1969, cuando ARPAnet estableció su primera conexión con tres Universidades de California y una en Utah.

Características:

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web, a través de Internet o de una Intranet. Las aplicaciones Web son populares, debido a la practicidad del navegador Web. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de clientes potenciales, es otra razón de su gran popularidad. Ejemplos de este tipo de aplicaciones son los Webmails, Wikis, Blogs, Tiendas en Línea (Online), entre otras.

Ventajas y Desventajas de la Programación Web:**Ventajas:**

- Se tiene un entorno centralizado.
- Al hacer actualizaciones a los sistemas, solo se tendrá que hacer en una computadora, en este caso, es el servidor.
- No se debe instalar ningún programa ejecutable en las máquinas cliente.
- Se tiene un mejor control sobre los sistemas en tiempo de ejecución.
- Se tienen actualizaciones constantes de los sistemas y de la información que muestran.

Desventajas:

- Los procesos son lentos.
- Se requiere de una conexión permanente a Internet.
- Existen deficiencias en cuanto a la seguridad de la información.
- Las versiones de los sistemas pueden no ser las óptimas para todos los navegadores o resoluciones de pantallas.

1.3 Programación de Aplicaciones de Escritorio:

¿Qué es una Aplicación de Escritorio?

Es un programa compilado, escrito en cualquier lenguaje, que le permite al usuario trabajar en una computadora.

Una aplicación de escritorio suele tener un único objetivo, por ejemplo: navegar en la Web, revisar correo, explorar el disco duro, etc. Una aplicación que posee múltiples programas se considera un paquete.

Las aplicaciones pueden tener distintas licencias de distribución como pueden ser las siguientes: freeware, shareware, trialware, etc. Todas las aplicaciones tienen un tipo de interfaz, que puede ser interfaz de texto o interfaz gráfica, inclusive puede tener ambas.

La principal diferencia entre las aplicaciones de escritorio a las aplicaciones Web, son que las primeras están alojadas directamente en la máquina que el usuario está utilizando y se consumen los recursos de dicha máquina, contrario a las aplicaciones Web, que se ejecutan y consumen recursos del servidor que las aloja.

Ventajas y Desventajas de las Aplicaciones de Escritorio:

Ventajas:

- Los procesos son rápidos.
- No requieren de una conexión a Internet, ya que en caso de que accedan a bases de datos en una Intranet, sólo se debe de configurar la computadora a la Intranet.
- Existe una mayor seguridad en los datos.
- Los sistemas no tienen problemas con las resoluciones de pantallas.
- Se tiene un mejor control sobre la máquina que ejecuta el sistema.

Desventajas:

- Se tiene un entorno descentralizado.
- Las actualizaciones se deben de instalar en cada una de las máquinas en donde se ejecuta el sistema.
- Se debe de instalar el sistema localmente.
- En caso de que el sistema falle, se pueden bloquear otras aplicaciones de la misma máquina.

- Las actualizaciones al sistema y a su información pueden ser más lentas.

1.4 Sistemas Distribuidos:

Definición de Sistema Distribuido:

Es una colección de elementos de cómputo autónomo que se encuentran físicamente separados, y que no comparten una memoria común, además de que se comunican entre sí, a través del intercambio de mensajes utilizando un medio de comunicación. En todo sistema distribuido se establecen una o varias comunicaciones que siguen un protocolo prefijado, mediante un esquema de “cliente-servidor”.

Características:

- Cada elemento de cómputo tiene su propia memoria y su propio Sistema Operativo.
- Control de recursos locales y remotos.
- Sistemas Abiertos (Facilidades de cambio y crecimiento).
- Plataforma no Estándar (Unix, Intel, etc.).
- Medios de comunicación (Redes, Protocolos. Dispositivos, etc.).
- Capacidad de Procesamiento en Paralelo.
- Dispersión y parcialidad.

Ventajas y Desventajas de los Sistemas Distribuidos:

Ventajas:

- Procesadores más poderosos y a menor costo.
- Desarrollo de estaciones con más capacidades.
- Las estaciones satisfacen las necesidades de los usuarios.
- Uso de nuevas interfaces.
- Disponibilidad de elementos de comunicación.
- Desarrollo de nuevas técnicas.
- Compartición de recursos (Hardware, Software).
- Eficiencia y flexibilidad (Respuesta rápida, ejecución concurrente de procesos, empleo de técnicas de procesamiento distribuido).
- Disponibilidad y confiabilidad.
- Mayores servicios que elevan la funcionalidad (Monitoreo, Correo Electrónico, etc.).
- Crecimiento Modular.

Desventajas:

- Requerimientos de mayores controles de procesamiento.
- Velocidad de propagación de información (muy lenta a veces).
- Servicios de replicación de datos y servicios con posibilidades de fallas.
- Mayores controles de acceso y proceso.

- Administraciones complejas.
- Costos altos (dependiendo de la infraestructura).

Modelo Cliente - Servidor:

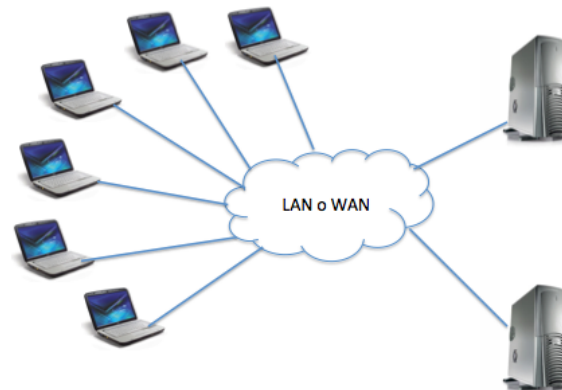


Figura 1.2: Modelo Cliente – Servidor (Lan o Wan)

En un Modelo Cliente-Servidor, se denomina Cliente a la máquina que solicita un determinado servicio, y se denomina Servidor a la máquina que lo proporciona.

Un Sistema Cliente-Servidor es aquel en el que uno de los Clientes y uno o más Servidores, que junto con un Sistema Operativo subyacente, y un Sistema de Comunicación entre procesos, forma un sistema compuesto que permite que el cómputo sea distribuido, hacer el análisis, así como la presentación de los datos.

El servicio puede ser la ejecución de un determinado algoritmo, el acceso a determinado banco de información, o el acceso a un dispositivo (Hardware).

Se puede aplicar el Modelo Cliente-Servidor dentro de una misma máquina, en donde el proceso Servidor y el proceso Cliente son dos procesos independientes que se ejecutan dentro de la misma instancia del Sistema Operativo.

Es importante la presencia de un medio físico de comunicación entre ambas máquinas, y será la naturaleza de este medio la que marque en muchos casos, la viabilidad del Sistema.

Se mencionan tres componentes básicos de software:

- Interacción con el usuario.
- Lógica de Aplicación.
- Bases de datos.

La interacción con el usuario se realiza básicamente en el servidor. El cliente hace de pasarela, del sistema de acceso a los elementos hardware que son el teclado y la pantalla. Los datos se envían sin formatear, y es el cliente el responsable de formatear los datos y realizar la función de interactuar con el usuario. En este caso, la aplicación y la base de datos se encuentran en el servidor.

Tanto la interacción con el usuario como la aplicación residen en el cliente, siendo el servidor en donde se almacenan los datos. Cliente-Servidor. La aplicación se distribuye en los tres niveles:

- Aplicación
- Datos
- Interfaces de usuario

Aplicaciones Distribuidas utilizando Dispositivos Móviles:

El diseño de aplicaciones modernas involucra la división de una aplicación en diferentes capas que son: la interfaz de usuario, la capa media de objetos de negocios y la capa de acceso a los datos. Es de gran utilidad, identificar los tipos de procesamiento que se puede esperar de una aplicación que realice, muchas aplicaciones pueden hacer las siguientes funciones:

- Cálculos.
- Validación de datos.
- Manipulación de datos.
- Interacción con otras aplicaciones externas o de servicios.
- Interacción con otros usuarios.

Con estos tipos de servicios, se puede hacer una clasificación en tres grupos, que a continuación se mencionan:

- Interfaz de usuario (Capa de Presentación):
 - Interacción con otros usuarios.
 - Interacción con aplicaciones externas.
- Procesos de negocios (Capa de Negocios):
 - Cálculos.
 - Validación de datos.
- Procesos de datos (Capa de Servicios de Datos):
 - Manipulación de datos.

La división de estos procesos de aplicaciones y su distribución entre diferentes procesos Cliente-Servidor, es conocido como Procesamiento Distribuido. En resumen, estos procesos que están dentro de estas tres categorías, es una distribución lógica y no refleja de forma necesaria, alguna opción de diseño físico sobre las computadoras, terminales u otros equipos. Es posible desarrollar una aplicación Cliente-Servidor distribuida basada sobre estas capas de Presentación, Lógica de Negocios y Servicios de Datos, para tener la aplicación entera ejecutándose sobre una simple computadora. De forma alterna, se puede esparcir estas tres capas, a través de un gran número de diferentes computadoras sobre una red, además de que se ha desarrollado una aplicación distribuida Cliente-Servidor de tres capas.

Con los Dispositivos Móviles:

Debido a que las aplicaciones distribuidas están teniendo cada vez un mayor auge en el mundo de la Informática, además de que es el Modelo más utilizado en la actualidad, es importante combinar este modelo de aplicaciones junto con los dispositivos móviles para que funcionen correctamente.

Aunque en la actualidad la programación para los dispositivos móviles está tomando mayor fuerza, todavía no es tan reconocido en la mayoría de las empresas para que desarrollen aplicaciones, o en su defecto, tengan en su tecnología desarrollos para dispositivos móviles, sin embargo, ya están empezando a considerar como una nueva opción para utilizarlo en su desempeño laboral. Esto no quiere decir que no existan aplicaciones para los dispositivos móviles en las empresas; de hecho, si existen aunque aún son limitadas, pues se tienen que considerar las funciones de la empresa, así como también, lo que se desea obtener.

CAPITULO No. 2: “HISTORIA DE LOS DISPOSITIVOS MOVILES”

Las calculadoras programables de Casio rivalizaban en prestaciones con las de Hewlett-Packard. Si la FX-750P podría considerarse como el buque insignia de la marca japonesa, la Serie HP48 lo era para los californianos. Las calculadoras de esta Serie, que se dividían en dos ramas, la S para los modelos estándar y la G para los de mayor funcionalidad, estuvo en producción entre los años 1990 y 2003. Las especificaciones comunes a todos los modelos de la gama eran una pantalla de 131 x 63 píxeles, un puerto de comunicaciones por infrarrojos y otro serie de 4 pines, y 512 Kb de Memoria. El modelo más alto de la gama, la HP 48GX, soportaba dos tarjetas de expansión de memoria, de forma que se podía llegar a acumular un total de 5 MB.

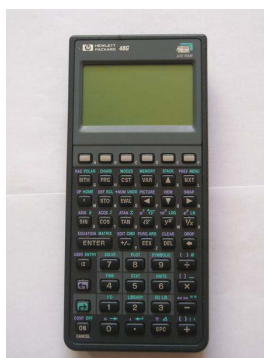


Figura 2.3: HP 48G

Tras la muerte de la Newton, nacieron los dos Dispositivos que durante unos años dominaron el mercado: la Pilot y el PocketPC.

Los Dispositivos de Palm se adelantaron a los de Microsoft en un par de años. En parte por ello, su salida a producción fue un éxito de ventas, llegando incluso a acumular, en 2001, un 60 por ciento del Mercado Americano. Sin embargo, Palm se resintió del declive global del Mercado de las PDA, comenzando un declive que coincidió prácticamente en el tiempo con la entrada en el Mercado del gigante de la Informática de Consumo: Microsoft.



Figura 2.4: Graffiti, el Sistema de Reconocimiento de Caracteres de la Palm OS,

En el año 2000 vio la luz el primer PocketPC, el hijo de la Newton, y que al contrario que ésta, si ha sobrevivido hasta el día de hoy, pese a no ser un éxito comercial de grandes dimensiones, gracias a la continuidad y el apoyo decidido de Microsoft.

Los primeros PocketPC tenían como Sistema Operativo, el llamado "Windows CE 3.0". Por su nombre parecía que era una versión aligerada del Sistema Operativo más utilizado en el mundo, pero en realidad no tenía mucha relación con éste.



Figura 2.5: Compaq iPAQ 3630, el primer Pocket PC de éxito

La mayor razón para el éxito del PocketPC ha sido su relativa sencillez de manejo y su integración con Computadoras de Escritorio basados en Windows. La integración es tal que para realizar una sincronización entre ambos basta con conectar el PocketPC a la Computadora con un cable.

Actualmente, las PocketPC y las Palm tienen pantallas de resolución VGA, en prácticamente todos los casos incorporan Protocolos de Comunicaciones Inalámbricos, como Bluetooth o Wi-Fi, o unidades de GPS. Por Prestaciones son, sin lugar a duda, los hermanos mayores del mundo de la Movilidad.

Pero si las PDAs son las líderes en lo referente a Prestaciones, en cuanto a unidades en el mercado, el rey indiscutible es el teléfono celular.

¿Quién no tiene un teléfono celular? En realidad, lo raro es que haya quien no tenga un teléfono totalmente funcional guardado en un cajón, porque se ha comprado otro, que en el plazo de unos meses, multiplicaba por diez o por cien las Prestaciones de su antiguo Aparato.



Figura 2.6: Motorola MicroTAC

Cada vez los móviles pueden hacer más cosas, y eso crea una doble necesidad. Por un lado, es necesario tener un Dispositivo que permita hacer esas cosas, sean las que sean (grabar video, escuchar audio en formato MP3, ver programas de televisión en tiempo real, etc.). Pero eso también provoca que cada vez se intenten hacer más cosas con el teléfono, lo que vuelve a hacer necesario un aparato más potente, que a su vez creará la necesidad de contenidos de más calidad, lo que a su vez, esto supone un factor importante: un círculo vicioso para el usuario pero que beneficia por igual a los Fabricantes y a las Compañías Telefónicas.

Sin embargo, los teléfonos celulares que verdaderamente pueden considerarse equivalentes a las PDAs son los de gama alta, o *Smartphones*.

El término "Smartphone" es engañoso. Su traducción literal sería "teléfonos inteligentes", y se utiliza para hacer referencia a cualquier teléfono de gama alta, englobando tanto a los Dispositivos de la Serie 60 o Superior de Symbian, como a los que funcionan bajo Windows Mobile o bajo Palm OS. De todas formas, la marca comercial *Smartphone* es propiedad de Microsoft.



Figura 2.7: Sony Ericsson P910i

Porque, efectivamente, hay una línea de teléfonos que funcionan bajo Windows Mobile. Por un lado, los PocketPC Phone Edition, que son híbridos de teléfono y PocketPC y por otro, los llamados comercialmente Smartphones. La diferencia entre ambos está, fundamentalmente, en la pantalla táctil. Los PocketPC Phone Edition tienen, mientras que los Smartphones no, y que se manejan de forma similar a los celulares Symbian de gama alta. Tanto PocketPC Phone Edition como PocketPC se engloban dentro de la gama de *Windows Mobile* de Microsoft.

También Palm mantiene su irreductible nicho de mercado dentro de los teléfonos de gama alta: la Serie Treo. Los Treo son híbridos de teléfonos y PDA, con teclado QWERTY, y funcionan, salvo en uno de los modelos (Treo 700w, que utiliza el PocketPC Phone Edition) bajo Palm OS.



Figura 2.8: Treo 700w

En todo caso, el mercado de los teléfonos de gama alta estaba muy interesado, hasta hace poco tiempo, por los modelos con el Sistema Symbian.

Sin embargo, en los últimos años hemos sido testigos de la llegada de uno de los representantes por excelencia de la década, que es el iPhone de Apple, Inc.. Si Apple fue la que abrió camino con la “Newton”, es el que ha vuelto a revolucionar el mercado con el iPhone, en parte porque ha sido el único fabricante que realmente ha conseguido desarrollar un dispositivo que integrara a todos los anteriores existentes por separado (teléfono, reproductor MP3 y PDA), reinventando la forma en la que se interactúa con él.

2.1 Smartphone:

Un “Smartphone” (Teléfono Inteligente) es un Dispositivo electrónico que funciona como un Teléfono Móvil con características similares a las de una Computadora. Es un elemento mitad Teléfono Móvil Clásico y mitad PDA, ya que permite hacer llamadas y enviar mensajes de texto como un Celular convencional. Una característica importante de casi todos los Teléfonos Inteligentes es que permiten la instalación de Programas para incrementar el Procesamiento de Datos y la Conectividad. Estas Aplicaciones pueden ser desarrolladas por el Fabricante del Dispositivo, por una Compañía Celular o por un tercero.

Los Teléfonos Inteligentes se distinguen por muchas características, entre las que destacan las Pantallas Táctiles, un Sistema Operativo, así como la Conectividad a Internet y el Acceso al Correo Electrónico.

2.2 PDA:

Un PDA es una Computadora de Mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Actualmente, estos Dispositivos pueden realizar muchas de las Funciones de una Computadora de Escritorio, pero con la ventaja de ser Portátil.

Al principio, los PDAs incluían Aplicaciones que estaban relacionadas con su función como agenda electrónica, es decir, se reducían a calendario, lista de contactos, bloc de notas y recordatorios. Con el paso del tiempo han ido evolucionando hasta los Dispositivos que existen actualmente, que ofrecen un amplio rango de Aplicaciones, como juegos, checar los correos electrónicos, inclusive la posibilidad de ver películas, crear documentos, navegar a través de Internet o reproducir archivos de audio en formato MP3, WMA, etc.

Las características de un PDA moderno son los siguientes: pantalla sensible al tacto, conexión a una Computadora para que la sincronización, ranura para tarjeta de memoria, y Puertos Infrarrojos, como Bluetooth o Wi-Fi.

2.3 Dispositivos Móviles en el Trabajo:

Los Handhelds se han convertido en equipos robustos para su utilización en la Gestión de Situaciones Empresariales, por ejemplo, para la Grabación de Información estando en la calle. Actualmente son utilizados para archivar una variedad de tareas, incrementando la eficiencia, como es la Digitalización de Notas, Gestión de Archivos, Capturas de Firmas, Gestión y Escaneo de partes de Código de Barras, etc.

Los Dispositivos Portátiles usados en el Trabajo, se han ido moldeando a través del tiempo una variedad de factores y formas, incluyendo los *Smartphones* en el extremo inferior, Dispositivos Portátiles, PDA, PC Ultra Móviles, Tablet PC (iPad), e incluso Laptops.

2.4. Sistemas Operativos para Dispositivos Móviles:

El Sistema Operativo destinado a ejecutarse en un Dispositivo Móvil necesita ser fiable y tener una gran estabilidad, ya que incidencias habituales y toleradas en Computadoras Personales como reinicios o caídas no tienen cabida en un Dispositivo de estas características. Además, ha de adaptarse adecuadamente a las consabidas limitaciones de memoria y procesamiento de datos, proporcionando una ejecución exacta y excepcionalmente rápida al usuario.

Estos Sistemas han de estar perfectamente probados y libres de errores antes de incorporarse definitivamente a la línea de producción. Las posibilidades que existen en una Computadora de realizar actualizaciones e incluso reinstalar mejores versiones del Sistema para cubrir fallas o deficiencias son más limitadas en un Dispositivo Móvil.

Es posible incluso que un aparato de esta naturaleza deba estar funcionando de manera interrumpida durante semanas e incluso meses antes de ser apagado y reiniciado a diferencia de lo que ocurre con una PC. El consumo de energía es otro tema importante: el Sistema Operativo debe de tener un uso más racional y provechoso posible de la batería, ya que esta limitada y el usuario siempre exige una mayor autonomía.

Todos estos aspectos de los Dispositivos Móviles, entre otros muchos, han de ser tomados en cuenta al momento de desarrollar un Sistema Operativo que sea competente en el Mercado, también que sea atractivo para los Fabricantes y que permita al Usuario sacar el máximo provecho de su Dispositivo.

Windows Phone:

Windows Phone, anteriormente llamado Windows Mobile, es un Sistema Operativo Móvil compacto desarrollado por Microsoft, y diseñado para su uso en *Smartphones* y otros Dispositivos Móviles. Windows Phone hace parte de los Sistemas Operativos con una Interfaz natural de Usuario NUI.

Se basa en el núcleo del Sistema Operativo Windows CE y cuenta con un conjunto de Aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de Escritorio de Windows. Además, existe una gran oferta de Software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de Windows Marketplace for Mobile.

Originalmente apareció bajo el nombre de *Pocket PC*, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas. Actualmente, la mayoría de los teléfonos con Windows Mobile vienen con un estilo digital, que se utiliza para introducir comandos pulsando en la pantalla. Windows Mobile ha evolucionado y cambiado de nombre varias veces durante su desarrollo, siendo la última versión llamada "Windows Phone 7", que fue anunciada el 15 de Febrero de 2010 y disponible a fines del 2010.

Symbian:

Symbian es un Sistema Operativo que fue producto de la alianza de varias Empresas de Telefonía Móvil, entre las que se encuentran Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDAs y Handhelds de PSION.

El objetivo de Symbian fue crear un Sistema Operativo para Dispositivos Móviles que pudiera competir con la Palm o el Windows Mobile de Microsoft, y ahora con el Android de Google, Inc., y el iOS de Apple, Inc.

C++:

El Comité para el Estándar ANSI C fue formado en 1983 con el objetivo de crear un lenguaje uniforme a partir del C original, desarrollado por Kernighan y Ritchie en 1972, en AT&T. Hasta entonces el Estándar lo marcaba el libro escrito en 1978 por estos dos Autores.

El Lenguaje C++ se comenzó a desarrollar en 1980. Su Autor fue B. Stroustrup, quien también trabajaba en AT&T. Al comienzo era una extensión del Lenguaje C que fue denominado "**C with Classes**". Este nuevo lenguaje comenzó a ser utilizado fuera de AT&T en 1983. El nombre **C++** es también de ese año, y hace referencia al carácter del Operador Incremento de C (++). Ante la gran difusión y éxito que iba obteniendo en el mundo de los programadores, AT&T comenzó a estandarizarlo internamente en 1987. En 1989 se formó un Comité ANSI para estandarizarlo a Nivel Americano e Internacional.

Actualmente, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales lo ha llevado a ocupar el primer lugar como Herramienta de Desarrollo de Aplicaciones. C++ mantiene las ventajas de C en cuanto a riqueza de Operadores y Expresiones, Flexibilidad, Concisión y Eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de **Java**, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar Aplicaciones en Internet.

Hay que mencionar que C++ ha influido en algunos puntos muy importantes del ANSI C, por ejemplo en la forma de declarar las funciones, en los punteros a void, etc. En efecto, aunque C++ es posterior a C, sus primeras versiones son anteriores al ANSI C, y algunas de las mejoras de éste fueron tomadas del C++.

J2ME:

J2ME es un subconjunto del Lenguaje Java. ¿Pero qué es el Lenguaje Java? Es un Lenguaje de Programación desarrollado a principios de los años Noventa por James Gosling y algunos de sus colegas de Sun Microsystems. Las Características principales de este lenguaje son las siguientes: que es de los llamados Orientados a Objetos, el cual permite desarrollar Aplicaciones más complejas con mayor facilidad, y también que fue diseñado para ser independiente de la Plataforma, lo que supuso que fuera una gran novedad en su momento.

Sin embargo, el caso de J2ME es especial, ya que es de todo, menos independiente de la Plataforma.

La Especificación J2ME está dividida en dos grandes grupos, dependiendo de la cantidad de funcionalidades para las que se quiera dar soporte. Esos Grupos son las llamadas Configuraciones. Hay dos Configuraciones, que dividen la Plataforma en dos grandes Grupos de Dispositivos.

En todo caso, J2ME es la Plataforma de desarrollo para celulares más extendida, porque no está limitada sólo a los teléfonos con el Sistema Symbian, porque durante mucho tiempo ha sido la única disponible para los desarrolladores, y porque algunos años en los que no ha sido la única, programar en J2ME era la única forma de implementar muchas funcionalidades. De hecho, hoy en día, sigue siendo la Plataforma más extendida, con mayor número de desarrolladores, y con mayor número de Aplicaciones disponibles.

Adobe Flash Lite:

Macromedia Flash Lite es una Aplicación Informática, se trata de una versión especial de Adobe Flash, Software de Adobe Systems. Fue creada para ser usada especialmente en teléfonos celulares y en otros dispositivos electrónicos portátiles pero no de telefonía, como Chumby e iRiver, permitiendo a sus usuarios acceder a contenido multimedia y Aplicaciones desarrolladas con Herramientas de Adobe Flash, que anteriormente sólo estaban disponibles desde una Computadora Personal.

iOS:

iOS (anteriormente se llamaba iPhone OS) es un Sistema Operativo Móvil de Apple, desarrollado originalmente para el iPhone, siendo después usado en el iPod Touch e iPad. Es un derivado de Mac OS X, que a su vez está basado en Darwin BSD. El iOS tiene 4 Capas de Abstracción: la Capa del Núcleo del Sistema Operativo, la Capa de “Servicios Principales”, la Capa de “Medios de Comunicación” y la Capa de “Cocoa Touch”. Todo el Sistema se encuentra en la partición “/root” del Dispositivo, ocupa un poco menos de 500 Megabytes.

Características:

Interfaz de Usuario:

La Interfaz de Usuario de iOS se basa en NUI con el concepto de manipulación mediante gestos multitáctiles. Los Elementos de la Interfaz se componen por Deslizadores, Interruptores y Botones. La Respuesta es inmediata y se provee de una Interfaz fluida. La interacción con el Sistema Operativo se realiza mediante Funciones como deslizar, tocar y pellizcar. Acelerómetros y Giroscopios internos son utilizados por algunas Aplicaciones para responder a movimientos y funciones, por ejemplo: sacudir el Dispositivo o Rotarlo.

Home Screen:

La pantalla principal (llamada "SpringBoard") es donde se ubican los íconos de las Aplicaciones y el Dock en la parte inferior de la pantalla donde se pueden anclar Aplicaciones de Uso Frecuente, aparece al desbloquear el Dispositivo o presionar el botón de inicio. La pantalla tiene una barra de estado en la parte superior para mostrar los datos, como la hora, el nivel de batería y la intensidad de la señal.

Tecnologías no Soportadas:

iOS no soporta Adobe Flash o Java, lo que afecta a los sitios web que utilizan estas tecnologías. Steve Jobs escribió una carta abierta donde critica a Flash por inseguro, con errores, así como consumir mucha batería, ser incompatible con Interfaces Multitouch e interferir con el servicio App Store. En cambio, iOS soporta HTML 5 como una alternativa a Flash.

Objective-C:

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

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

A diferencia de otros lenguajes de las GCC, Objective-C no ha sido estandarizado por ningún Organismo Internacional, sino que fue NeXTSTEP, y ahora Mac OS X quienes han contribuido a crear este lenguaje. NeXT, la Empresa que creó NeXTSTEP cedió la implementación de Objective-C a las GCC, y ambos están ahora trabajando en mantener y mejorar el lenguaje.

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

- Sentencias de control de flujo.
- Los tipos de datos fundamentales, estructurales y punteros.
- Conversiones implícitas y explícitas entre tipos.
- Los ámbitos de las variables: Globales, estáticas y locales.
- Las funciones y su sintaxis.

- Las directivas del preprocesador, aunque cabe señalar que Objective-C añade más directivas del preprocesador, y también añade las llamadas directivas del compilador.

Lenguaje Marcadamente Dinámico:

Si hubiera que elegir una característica que diferencia a Objective-C de otros lenguajes, ésta sería que es un lenguaje muy dinámico, en el sentido de que muchas decisiones que muchos lenguajes toman en tiempo de compilación, Objective-C les deja para el tiempo de ejecución.

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

Los Cinco Tipos de Dinamismo que diferencian a Objective-C de otros lenguajes son:

1. Memoria dinámica.
2. Tipos dinámicos.
3. Introspección.
4. Enlace dinámico.
5. Carga dinámica.

CAPITULO No. 3: “APPLE, INC.”

3.1 Historia de Apple, Inc.:



Figura 3.1: Logotipo de Apple, Inc.

Apple, Inc. es una Empresa estadounidense de Tecnología Informática. Fue fundada en 1976, tuvo un rol prominente en la Revolución Informática de los años setenta con su Computadora Apple II. Sus inicios son poco menos que curiosos, y forman parte del Folklore Informático. Tanto Steve Jobs como Steve Wozniak, eran dos grandes aficionados a la Electrónica en general, y a la Informática en particular, en especial Wozniak. Éste, desde pequeño, dedicaba sus ratos libres a idear circuitos electrónicos complejos, para después optimizarlos al máximo. De esta afición, surgió su primera computadora, la Apple I, la cual fue presentada en la “Homebrew Computer Club”. Aunque para Wozniak, la Construcción de Computadoras no era más que un simple hobby, Jobs rápidamente vio la oportunidad de Negocio, por lo que empezó a promocionar la Computadora entre otros aficionados de la Informática del “Homebrew Computer Club” y otras Tiendas de Electrónica Digital.

Pronto empezaron pequeños encargos que resolvían ellos mismos con máquinas hechas a mano, llegando a vender unos 200 Apple I. Se les unieron más amigos, pero las prestaciones del Apple I eran limitadas, así que empezaron a buscar Financiamiento, Jobs, conoció a Mike Markkula, que accedió a invertir, creándose así el 1º de Abril de 1976 Apple Computer.

Con estos nuevos ingresos, diseñaron el Apple II, marcando un hito en su momento y definiendo lo que debía ser una computadora personal en aquella época. Cuando la Compañía tuvo acceso al público en 1980, ganaron millones de dólares en días con la Apple II Family. Se desarrolló el Apple III, pero en 1983, los diseñadores estaban pensando en una revolucionaria forma de computadora, un nuevo concepto, una computadora con una Interfaz Gráfica, Mouse e íconos; la computadora como la conocemos hoy en día.

La Empresa se dio cuenta de que el futuro estaba en la creación de una interfaz gráfica más sencilla y accesible para el usuario, es decir, las interfaces gráficas que poco a poco empezaban a salir de los Laboratorios para intentar colarse en Computadoras Comerciales.

Tras la salida del primer IBM PC en 1981, Apple comprendió que necesitaba un golpe de efecto para recuperar la delantera en el mundo de la Informática Personal, y su gran apuesta y esperanza era el Apple Lisa, la segunda Computadora Comercial salida al mercado después del Xerox Star.

Apple I:

Para entonces, un Apple I consistía de una placa con diversos componentes soldados a la misma y una serie de puertos que interaccionaban con la computadora.

El éxito fue tal, que no pudieron dar abasto con tanta demanda. Las Características del Apple I eran limitadas por el poco dinero del que disponían Jobs y Wozniak.



Figura 3.2: Apple I.

El Apple II y el Éxito:

Con el dinero ganado, gracias a la venta del Apple I, en Apple se pudo comenzar a pensar en una máquina más ambiciosa: el Apple II. Para Wozniak, el Apple II iba a ser simplemente la computadora que hubiera querido construir, si no hubiera tenido las limitaciones económicas que tuvo con el Apple I. Tuvo la idea de incluir en el equipo memoria de vídeo para administrar la visualización en color, del mismo modo que incluía numerosas tarjetas de expansión de modo que los usuarios pudieran ampliar las capacidades de la computadora, según sus propias necesidades.

Con el diseño del primer prototipo del Apple II recién terminado, Apple decidió asistir a la "Personal Computing Festival", un nuevo Certamen, la incipiente Industria de la Microinformática podía mostrar sus progresos, así como hacer Negocio y Contactos. En el vuelo a Filadelfia, lugar en donde se celebrara en 1976, Apple coincidió en el avión con los componentes de Processo Technology y su recién diseñada computadora Sol, coincidencia que tuvo un gran impacto en el futuro inmediato de Apple.

La gran mayoría de máquinas y Empresas dedicadas a la Microinformática en 1976, y por consiguiente, prácticamente todas las que formaban el Festival, eran en general jóvenes hackers con kits que los usuarios debían montar y/o soldar para poder hacer funcionar, por lo tanto enfocadas sobre todo a los entusiastas de la Electrónica. Sin embargo, Processor Technology ofrecía una imagen mucho más seria y profesional, y su Sol era

una Computadora que si bien estaba enfocado, al igual que los demás, como un kit que debía montarse, también se daba la posibilidad de adquirirla como Computadora ya montada y lista para utilizar.

Fue entonces cuando Steve Jobs comprendió que el futuro no estaba en placas con componentes que los usuarios debieran montar y soldar ellos mismos, sino en computadoras como el Sol, ya montadas y preparadas para el uso y disfrute, según salían de la caja. Por lo tanto, el Apple II, el cual aún no había salido a la venta aún, debía incorporar la salida de vídeo, el teclado y todo aquello que necesitara en una caja anodina de plástico para hacer que su uso fuera más sencillo.

Mejores Sistemas de Alimentación Eléctrica y Control del Calentamiento del Equipo, siendo el Apple II una computadora completamente silenciosa, gracias a que no necesitaba ventiladores para controlar su temperatura.

El Modelo Final de Apple II se presentó al público en Abril de 1977, convirtiéndose en el canon de lo que debía ser a partir de entonces una computadora personal. Por este motivo, Apple cambió su logotipo por el famoso logo de la manzana coloreada, que recordaba que el Apple II era una de las primeras computadoras en disponer de monitor a color. A mediados de 1979, Apple presentó el Apple II+, una evolución del Apple II dotada en particular, con más memoria (48 Kb ampliable a 64 Kb) y lenguaje de programación BASIC.

Gran parte del éxito de Apple se basaba en que fue una computadora realizada para gustar tanto a los expertos hackers como al público más profano en Términos de Informática. Pronto, la base de software para el Apple II comenzó a crecer, haciendo más atractivo la computadora al resto de público, en especial cuando apareció en el mercado la primera hoja de cálculo de la historia, la VisiCalc, la cual hizo vender ella sola miles de computadoras Apple II.



Figura 3.3: Comparación de los Equipos Apple II, Apple III y la Apple Lisa.

Mientras que el Apple II conocía un éxito creciente, Apple ya trabajaba en sus sucesores. Sara debía ser una evolución del Apple II, una máquina de transición antes de Lisa y Macintosh que serían una nueva clase de equipos. Para el sucesor del Apple II, Steve Jobs quería una máquina más avanzada, aún para contribuir al mercado de la Informática de Empresa.

Sara fue presentada finalmente en Mayo de 1980 bajo el nombre de Apple III. Desgraciadamente, algunas elecciones técnicas, entre las cuales se encontraba la no incorporación de un ventilador, fueron la razón de que muchos equipos se sobrecalentaran, incluso antes de suministrarse. Miles de Apple III tuvieron que ser sustituidas.

Algunos meses más tarde, en Noviembre de 1981, Apple lanzó una nueva versión del Apple III, que corregía todos los grandes problemas de la versión inicial. Un Apple III+ salió incluso en 1983. Pero los problemas de sus comienzos desalentaron a los compradores y fueron la razón de que el Apple III se convirtiera en el primer gran fracaso comercial de Apple. Sólo se habían vendido 65,000 equipos a final del verano, mientras que Apple pensaba vender millones como el Apple II. Los equipos que salieron en los años siguientes, llevaban el nombre de Apple II, para olvidar la mala racha del Apple III.

Pese al Fracaso del Apple III, Apple aún tenía dos opciones: la Lisa y el Macintosh. La gran apuesta de Apple era Lisa, la cual debía convertirse en la nueva Generación de Computadoras de Apple y con el que se pretendía también atacar el Mercado Empresarial que se le había negado con el Apple III, y que sin embargo estaba siendo ganado por el IBM PC, mientras que el Macintosh era un Proyecto iniciado por Jeff Raskin para construir una pequeña Computadora muy económica y fácil de usar pensado para el Mercado Doméstico.

Apple Lisa:

Apple Lisa estaba pensado para ser la nueva Generación de Computadoras, y para conseguirlo no se repararon en gastos. Se incluyó en el Modelo Final un monitor, dos disqueteras, un disco duro de 5 Megabytes (algo inmenso para esa época) y todo un inmenso Megabyte de RAM. Sin embargo, lo que primero saltaba a la vista no eran todas estas características, sino su Novedosa Interfaz de Usuario, el cual estaba basado en Íconos que se activaban apuntando con una flecha controlada por un dispositivo curioso, llamado Mouse.

El 12 de Diciembre de 1980, Apple entró a cotizar en la Bolsa. Hasta ese momento, sólo algunos de los empleados de Apple poseían Acciones de la Empresa. En unos minutos, las 4,600,000 millones de Acciones se vendieron a 22 dólares la unidad, aumentando instantáneamente en \$100,000,000 de dólares el Capital de Apple. Al mismo tiempo decenas de empleados de Apple se convirtieron en millonarios.

El 12 de Agosto salía al mercado la principal amenaza a la Hegemonía de Apple, el IBM PC. Pese a que no era una máquina muy innovadora, la imagen de marca de IBM le otorgaba un gran atractivo en el Sector Empresarial, por lo que se convirtió en un fulgurante éxito.

El Macintosh:

Tras el fracaso del Apple III y el Apple Lisa, la Compañía se volcó en el pequeño proyecto de Jeff Raskin primero y Steve Jobs después: el Macintosh. Tras numerosos retrasos y problemas, especialmente con el software, el cual no estuvo preparado hasta prácticamente el último momento, y tras un arriesgada Campaña de Publicidad, la Computadora fue presentada el 24 de Enero de 1984.

Las previsiones de venta iniciales de medio millón de unidades vendidas hasta fin de año parecían que podrían cumplirse durante los primeros meses de 1984, pero poco a poco las ventas se fueron lentas, provocando el pánico en Apple.

Entre los muchos motivos que hicieron erosionarse las ventas del Macintosh, se puede destacar el alto precio (\$2,495.00 USD), la escasa Memoria RAM instalada en la Computadora (sólo 128 Kbytes), la inclusión de una única unidad de disco, la no inclusión de puertos de expansión y la falta de software, ya que salvo Apple sólo Microsoft acudió a la cita del Mac con productos de software.



Figura 3.4: Macintosh.

MacBook Pro:

La MacBook Pro es la primer Computadora portátil de Apple Inc., en soportar un Procesador Intel y fue presentado a la prensa en la Feria Tecnológica MacWorld 2006, el 10 de Enero de 2006. La MacBook Pro es el sucesor del Modelo PowerBook G4, una Computadora Portátil Profesional de la Compañía que ha mantenido el Procesador PowerPC G4 por 3 Generaciones.

La nueva MacBook Pro es 4 veces más rápido que la última PowerBook G4. La MacBook Pro lleva dentro un microprocesador Intel Core 2 Duo de hasta 2.6 GHz. Además incorpora una videocámara iSight y la Aplicación de Contenidos Digitales "Front Row", originalmente introducidos para el Modelo iMac.

Además incorpora las últimas actualizaciones del PowerBook G4, como un teclado retroiluminado, sensor de iluminación para ajustar automáticamente el brillo de la pantalla y el teclado, la pantalla de resolución XVGA (VGA Extra) y además tiene nuevas Funciones como el MagSafe, que magnéticamente desconecta el cable de electricidad cuando este es arrancado o separado, evistando que se pueda caer la MacBook Pro. Todos las MacBook Pro incluyen sensores de movimiento que permiten detener el disco duro en caso de caída, evitando así una pérdida de datos.



Figura 3.5: MacBook Pro.

3.2. Historia del Mac OS:

El 24 de enero de 1984, Apple Computer Inc., introdujo la computadora personal Macintosh, con el modelo “Macintosh 128K”, el cual incluía el Sistema Operativo Mac OS, que era conocido en esa época como “System Software”.

A las computadoras Macintosh, se les atribuye con frecuencia la popularización de las interfaces gráficas de usuario. El Mac OS ha sido pre-instalado en casi todas las computadoras Macintosh vendidas. El sistema operativo también se vende por separado en las Tiendas Apple y en línea. El Mac OS original estaba basado de forma parcial en el Lisa OS, que previamente era comercializado por Apple para la computadora Lisa en 1983, y también usaba conceptos del Xerox PARC Xerox Alto, el cual Steve Jobs y otros miembros del equipo Macintosh habían visto.

Lanzamiento:

La primera versión del Mac OS, llamado “System”, se puede distinguir de una manera más fácil de los otros sistemas operativos de la misma época, ya que no usa una interface de línea de comandos, además de que fue uno de los primeros sistemas operativos en usar una completa interfaz de usuario. Adicionalmente al núcleo del sistema está el “Finder”, que es una aplicación que se utiliza para administrar archivos, la cual también mostraba el escritorio. Los dos archivos estaban contenidos en una carpeta etiquetada como “System Folder”, en la cual contenía otros archivos necesarios, como el controlador de la impresora, que eran necesarios para interactuar con el System.

System 1, 2, 3 y 4:

Estas ediciones sólo podían correr una aplicación a la vez, aunque existían algunas aplicaciones especiales como el “Servant”, “MultiMac” o el “Switcher”, podían evitar esto en cierta medida. Los System 1.0, 1.1 y 2.0 utilizaban un sistema de archivos con un solo nivel de directorios, llamado Macintosh File System; su soporte para carpetas era incompleto. El System 2.0 agregó soporte para AppleTalk para usar el recientemente introducido “LaserWriter”. El System 2.1 (Finder 5.0) implementaba el HFS, o mejor conocido como el sistema de archivos jerárquicos, que tenía directorios reales. Esta versión era específica para el Hard Disk 20 y sólo implementaba el HFS en la RAM, el arranque y muchos de los diskettes mantuvieron el volumen MFS de 400K. El System 3.0 fue introducido con el Mac Plus, que fue implementado de manera oficial el HFS y las unidades de arranque de 800K, agregando soporte para varias tecnologías nuevas como el SCSI y el AppleShare, y la Papelera “inflable”. El System 4.0 venía con el Mac SE y el Macintosh II, los cuales requerían soporte adicional para los primeros zócalos de expansión, el Apple Desktop Bus (ADB), discos rígidos internos en el Mac II, color, monitores grandes y el primer procesador Motorola 68020.

System Software 5:

El System 5 agregó el MultiFinder, una extensión que permitía al sistema correr varios programas al mismo tiempo. El sistema usaba el modelo de multitarea cooperativa, en el sentido de que daba tiempo a las aplicaciones en segundo plano sólo cuando la aplicación que se ejecuta daba el control. Un cambio inteligente en las funciones del sistema fue que las aplicaciones que eran llamadas para atender eventos hacían que las aplicaciones existentes compartieran el tiempo automáticamente. Los usuarios podían optar por no usar el MultiFinder, y por lo tanto, fijarlo con una sola aplicación por vez, como era en las versiones anteriores del System Software.

El System Software 5 fue el primer sistema operativo de Macintosh en tener número de versión unificada del “Macintosh System Software”, al contrario que los números usados para el System y el Finder.

System Software 6:

El System Software 6 fue una versión consolidada del Mac OS, produciendo un sistema operativo completo, estable y de larga duración. Las dos introducciones mayores en el hardware requirieron de soporte adicional en el System 6, en donde el procesador Motorola 68030 y el SuperDrive de 1.44 MB aparecieron con el Macintosh IIfx y el Macintosh SE/30. Más tarde, incluyó soporte para las características de la primera laptop especializada, con la introducción de la Macintosh Portátil. A partir del System 6, el Finder tendría un número de versión unificado, casi coincidiendo con el del System, para que no exista la confusión originada por las diferencias entre los System anteriores.

System 7:

El 13 de mayo de 1991 fue lanzado el System 7. Fue la segunda mayor actualización del Mac OS, agregando una gran renovación a la interfaz gráfica de usuario, nuevas aplicaciones, así como mejoras en la estabilidad y muchas características nuevas. Su introducción coincidió con lanzamiento y el soporte de la línea Macintosh con 68040.

La característica más importante del System 7 es el soporte para intercambio de página, el cual previamente estaba sólo disponible como un agregado de terceras compañías.

La Papelera, bajo el System 6 y anteriores, se vaciaba por si misma automáticamente cuando se apagaba la computadora o, si el MultiFinder no estaba corriendo, cuando se lanzaba una aplicación. El System 7 reimplementaba la Papelera como una carpeta especial oculta, que permitía mantener los archivos entre re-inicios del sistema, hasta que el usuario de forma deliberada ejecutaba el comando "Vaciar la Papelera". System 7.1 fue principalmente una versión para corregir errores, con características menores que fueron agregadas, además de que no sólo fue el primer sistema operativo que se tenía que comprar, sino también en recibir una función llamada "Pro", que tenía algunas características extra. El System 7.1.2 fue el primero en soportar Macs basadas en PowerPC. El System 7.1 también introdujo el System Enablers, o habilitadores del sistema, como un método para soportar nuevos modelos sin actualizar los archivos del System en uso, además esto agregó archivos extras en la carpeta del sistema que algunos usuarios consideraban no muy grato.

El System 7.5 introdujo varias características consideradas como de "alto nivel", ya que algunos usuarios opinaban que deberían de haber estado mejor elaboradas. Varias de las nuevas características estaban basadas en aplicaciones shareware que Apple compró e incluyó en el nuevo sistema. En las nuevas máquinas PowerPC, el System 7.5 tenía diversos problemas de estabilidad, debido parcialmente a un nuevo administrador de memoria, y al pobre manejo de errores del sistema operativo en el código PowerPC. Estos problemas no afectaban a las máquinas que tenían la arquitectura 68.000.

Mac OS 7.6:

En esta versión, se mejoró la estabilidad del Mac OS 7.6 en las Mac con PowerPC, el cual dejó el sobrenombre de "System" por un nombre que estaba más adecuado a la Mercadotecnia para que se pudieran otorgar las Licencias en un mercado de clones de Macintosh de otros fabricantes. Mac OS 7.6 necesitaba un CPU 68030 y ROM de 32 bits limpia, y se discontinuaba el soporte para muchas de las primeras Macs, incluyendo la Mac Plus y la Mac II.

Mac OS 8:

El Mac OS 8 fue lanzado el 26 de julio de 1997, poco después que Steve Jobs regresara a la compañía. Se lanzó principalmente para mantener avanzando al Mac OS en un momento difícil para Apple. Originalmente estaba planeado como Mac OS 7.7, fue nombrado como "8" para aprovechar una laguna legal y cumplir con la meta de Jobs de terminar con las Licencias de terceros fabricantes para el System 7 y eliminar el mercado de clones Macintosh. El 8.0 agregaba varias características del cancelado proyecto Copland, mientras que el sistema operativo básico quedaba sin cambios. Además, se incluyó un Finder multi-hilos, permitiendo una mejor multitarea. La interfaz gráfica de usuario fue cambiada de apariencia con un nuevo aspecto de sombreado en escala de grises llamado Platinum, y se agregó la posibilidad de cambiar los temas de apariencia al Panel de Control. Esta capacidad fue proporcionada por una nueva capa API de "apariciencia", uno de los pocos cambios importantes en el sistema operativo.

El Mac OS 8.1 vio la introducción de una versión mejorada del sistema jerárquico de archivos llamado HFS Plus, el cual solucionaba muchas de las limitaciones de los primeros sistemas. Hubo otros cambios en la interface, como la separación de las funciones de red de la impresión, y algunas mejoras en la conmutación de tareas. Sin embargo, en los aspectos técnicos de base, el Mac OS 8 no era muy diferente al System 7.

El Mac OS 8.5 se centró en la velocidad y estabilidad, por mucho del antiguo código 68k, el cual fue reemplazado por el código nativo para el PowerPC, así como también mejoró la apariencia del sistema, aunque la función de tematización se redujo a finales del desarrollo.

Mac OS 9:

El Mac OS 9 fue lanzado el 23 de octubre de 1999. Las primeras ediciones del Mac OS 9 fueron numeradas como 8.7. El Mac OS 9 agregó soporte mejorado para la red inalámbrica AirPort. Introdujo una temprana implementación de soporte multi-usuario. Una máquina de búsqueda Sherlock mejorada era uno de los nuevos complementos. Mac OS 9 también proveía una implementación y administración de memoria bastante mejorada. AppleScript fue ampliado para poder controlar las redes y el protocolo TCP/IP.

El Mac OS 9 también fue el primero en usar el Apple Software Update, centralizado para encontrar e instalar actualizaciones del Sistema Operativo y del Hardware. Otras características incluyen software de cifrado de archivos en tiempo real con tecnología de código de firma y Keychain, paquetes de Remote Networking and File Server, y una lista mejorada de controladores USB.

El Mac OS 9 también agregó algunas tecnologías de transición para ayudar a los desarrolladores de aplicaciones a adaptar algunas características del Mac OS X, antes de introducir el nuevo Sistema Operativo al público, facilitando la transición. Éstas incluían nuevas APIs para el archivo del sistema, la integración de la librería Carbon para poder enlazar las aplicaciones en lugar de las API tradicionales, las aplicaciones adaptadas de esta forma podían ejecutarse en forma nativa en el Mac OS X. Otros cambios que se hicieron en el Mac OS 9 para permitir que sea arrancado en el “entorno clásico” dentro del Mac OS X.

A continuación se mencionan las Versiones del Mac OS 9:

- Mac OS 9.0
- Mac OS 9.0.2
- Mac OS 9.0.3
- Mac OS 9.0.4
- Mac OS 9.1
- Mac OS 9.2
- Mac OS 9.2.1
- Mac OS 9.2.2

Mac OS X:

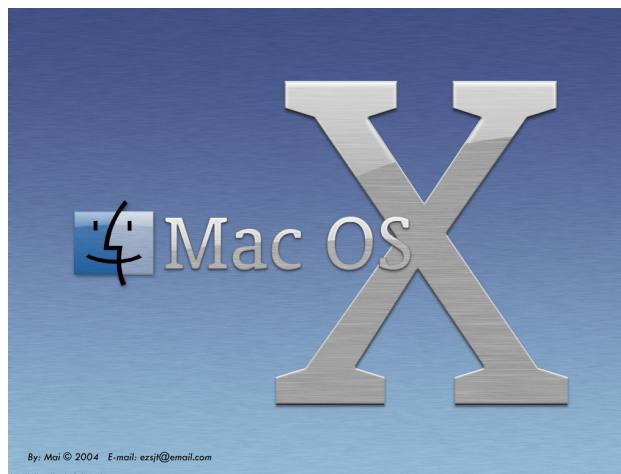


Figura 3.6: Sistema Operativo Mac OS X.

El Mac OS X es la línea de sistemas operativos gráficos desarrollados, promocionados y vendidos por Apple, Inc., el cual sucedió al original Mac OS, que fue el principal sistema operativo de Apple desde 1984. Al contrario que el anterior sistema operativo de Macintosh, el Mac OS X es un sistema operativo que está basado en Unix, construido con tecnología desarrollada en NeXT, en la segunda mitad de la década de 1980 hasta principios de 1997, cuando Apple compró la compañía.

La primera versión fue el Mac OS X Server 1.0 en 1999, el cual mantenía bastante de la apariencia tipo “Platinum” que tenía el anterior sistema operativo de Mac, e incluso llegó a parecerse al OPENSTEP en algunas partes. La versión orientada a escritorio, el Mac OS X 10.0, surgió en marzo de 2001 con la nueva interfaz de usuario llamada “Aqua”. Desde entonces, han aparecido otras ocho versiones distintas para los usuarios finales, así como también para los servidores, la más reciente es el Mac OS X 10.8, en julio de 2012. Las versiones del Mac OS X llevan los nombres de los grandes felinos. Por ejemplo, Apple llama al Mac OS X 10.5 “Leopard”, mientras que la versión previa se llamó “Tiger”.

A continuación, se mencionan las versiones del Mac OS X:

- Mac OS X 10.0 (Cheetah)

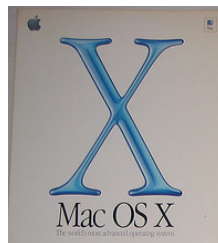


Figura 3.7: Mac OS X 10.0 (Cheetah).

- Mac OS X 10.1 (Puma)

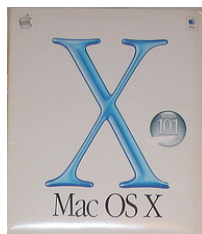


Figura 3.8: Mac OS X 10.1 (Puma).

- Mac OS X 10.2 (Jaguar)

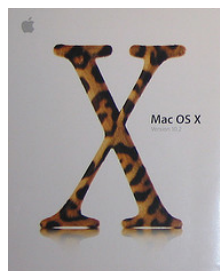


Figura 3.9: Mac OS X 10.2 (Jaguar).

- Mac OS X 10.3 (Panther)



Figura 3.10: Mac OS X 10.3 (Panther).

- Mac OS X 10.4 (Tiger)



Figura 3.11: Mac OS X 10.4 (Tiger).

- Mac OS X 10.5 (Leopard)



Figura 3.12: Mac OS X 10.5 (Leopard).

- Mac OS X 10.6 (Snow Leopard)



Figura 3.13: Mac OS X 10.6 (Snow Leopard).

- Mac OS X 10.7 (Lion)



Figura 3.14: Mac OS X 10.7 (Lion)

- Mac OS X 10.8 (Mountain Lion)



Figura 3.15: Mac OS X 10.8 (Mountain Lion)

**CAPITULO No. 4: “PLATAFORMA DE
DESARROLLO DEL iPhone Y EL
iPad”**

4.1 Xcode 4:



Figura 4.1: Ícono de Xcode.

Xcode es el conjunto de herramientas completo para la creación de aplicaciones de OS X y el iOS, y con Xcode 4, las herramientas se han rediseñado para ser más rápido, fácil de usar y útil que nunca. El IDE de Xcode entiende el proyecto en cada detalle, identifica errores tanto en la sintaxis y la lógica, e incluso corregir el código para uno, simplemente, Xcode 4 ayudará a escribir mejor el código.

Xcode 4 tiene una nueva interfaz de usuario, construido sobre tecnologías probadas que la propia Apple utiliza para construir OS X y el iOS, y que se han producido más de 250,000 aplicaciones para OS X y el iOS.

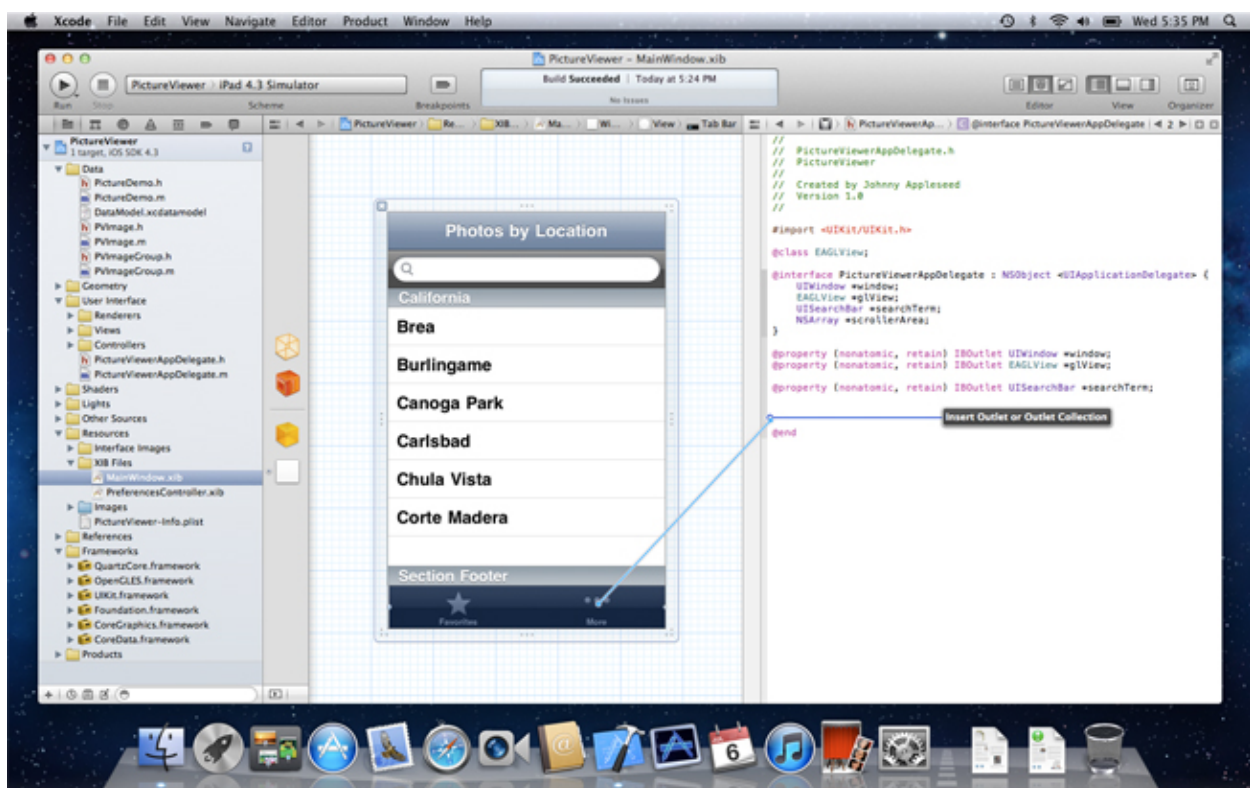


Figura 4.2: Interfaz de Usuario de Xcode 4.

Ventana Única:

Como usuario experimentado de Xcode, lo primero que notará de Xcode 4 son las numerosas ventanas utilizadas para realizar las tareas de desarrollo que trabaja todos los días se han consolidado en una sola ventana. El área de trabajo del Xcode 4 tiene varios elementos de interfaz de usuario únicos que hacen que sea fácil para trabajar en diversas tareas, incluso en proyectos múltiples, sin estorbar su área de trabajo. Su editor es siempre frontal y central.

Navegadores:

En el lado izquierdo hay una colección de los navegadores, incluyendo la lista de los archivos del proyecto, los símbolos ordenados, una interfaz de búsqueda central, seguimiento de problemas, los datos de depuración con seguimiento de pila compresibles, puntos de interrupción activos e inactivos, y una colección permanente de registros. La interfaz de usuario unificada del navegador ofrece en directo el filtrado del contenido y los resultados de la búsqueda, para que pueda centrarse en la tarea actual.

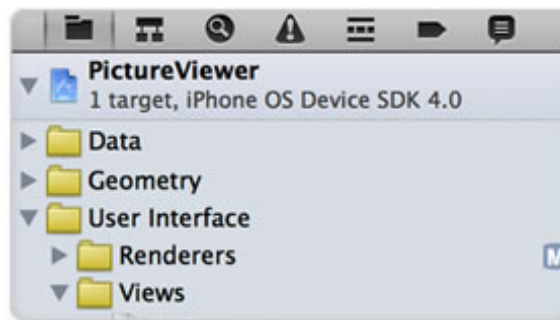


Figura 4.3: Navegadores.

Barra de Salto:

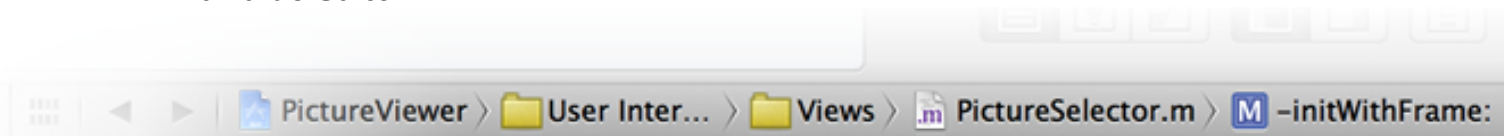


Figura 4.4: *Jump Bar* (Barra de Salto).

En la parte superior de cada panel del editor es una barra de ruta que muestra la ubicación relativa del archivo actual. Haciendo clic en cualquier punto de la ruta para pasar de inmediato a cualquier otro archivo en ese nivel. Esta es la Barra de Salto ("*Jump Bar*"), y es tan eficiente que se podría dedicar todo el escritorio al código fuente, rápidamente saltando de un archivo a otro.

Interface Builder se encuentra Integrado:

En Xcode 4, Interface Builder ha sido completamente integrado con el IDE de Xcode, en el cual no existe ninguna solicitud por separado.

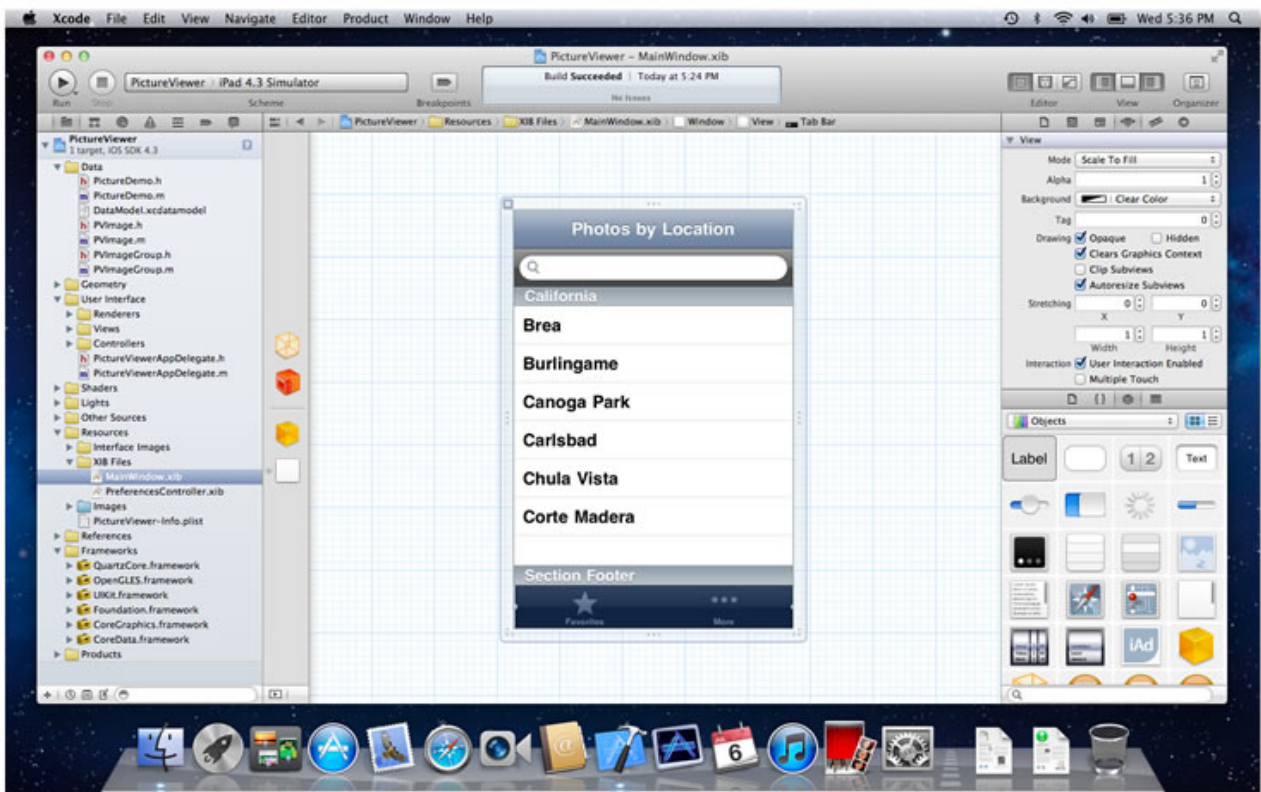


Figura 4.5: Interface Builder se encuentra integrado dentro de Xcode 4.

Cuando se selecciona un archivo de la interfaz gráfica (*.nib/.xib*) del proyecto, se abrirá el editor de Interface Builder dentro de Xcode. Del lado derecho, se abre la zona de Utilidad, en la cual se muestra todo el complemento de los inspectores de la interfaz, así como la biblioteca de controles y objetos de la interfaz de usuario. Arrastrando un control de la biblioteca, y al dejar caer sobre el lienzo, a la disposición del OS X o de la aplicación iOS.

Esta es la mejor parte: se pueden arrastrar las conexiones directamente desde el diseño de la interfaz de usuario hacia el código fuente. El nuevo diseño del editor de Xcode 4 es una división que permite que sea más fácil de realizar las conexiones de las acciones y los outlets, simplemente arrastrando la conexión con el código existente (*un solo gesto y, ¡ya está!*). ¿Todavía no se tiene el código listo para conectarse? Xcode creará un nuevo outlet o acción para uno, sólo se tiene que arrastrar hacia un espacio vacío en el archivo de origen y Xcode genera el código enseguida.

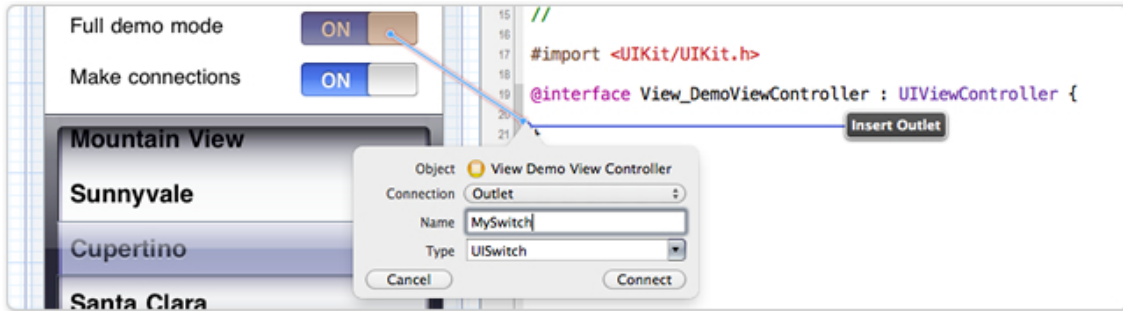


Figura 4.6: Arrastre de las conexiones desde la Interfaz de Usuario hacia el Código Fuente.

Asistente:

Al momento de estar escribiendo el código fuente, o diseñando la interfaz gráfica, no se produce el cambio de forma aislada. Incluso el más pequeño detalle puede tener un efecto en cascada a lo largo del proyecto. A menudo, se necesita ver más que el archivo actual, también hay que revisar todos los documentos que estén relacionados con él.

Con el nuevo “Asistente” de Xcode, el diseño del editor de dos paneles en Xcode 4 es más potente. Al momento de activar el “Asistente”, el IDE anticipa que otros archivos se necesitan para ver, a medida de cómo se va trabajando. ¿Se tiene que editar una nueva clase derivada? El “Asistente” le mostrará el código de la clase que se hereda. ¿Se tiene que escribir el código de implementación de nueva cuenta? El “Asistente” muestra automáticamente el encabezado correspondiente. En el diseño de la interfaz gráfica, el “Asistente” muestra el controlador adecuado, por lo que arrastrar y soltar las conexiones del código es muy simple. El diseño del Modelo de datos abrirá las clases que respaldan sus Modelos (*todo de forma automática*).

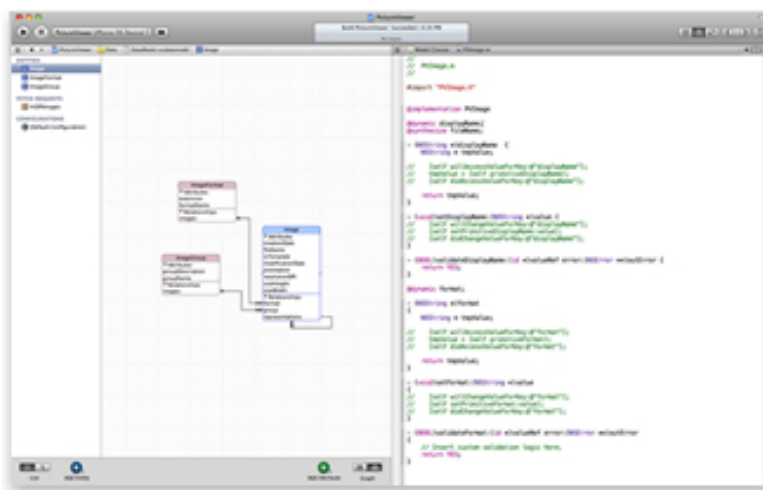


Figura 4.7: Asistente de Xcode.

Compilador Apple LLVM:

El Apple LLVM es el compilador con tecnología de próxima generación que alimenta a Xcode 4. Con base en el vibrante código abierto LLVM.org, el proyecto liderado por los ingenieros de Apple, el compilador Apple LLVM es el pensamiento moderno, dirigido hacia el iPhone, el iPad y la Mac multi-núcleo.

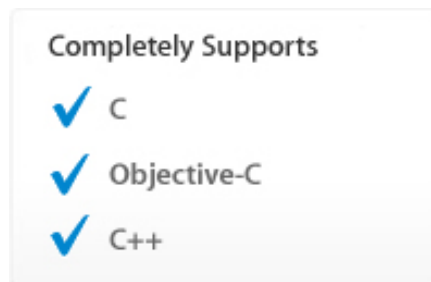


Figura 4.8: El Apple LLVM tiene un soporte completo para C, Objective-C y C++.

El Apple LLVM es rápido. Compila el código dos veces tan rápido como el GCC (*GNU Compiler Collection – Colección de Compiladores GNU* -), sin embargo, también produce que se ejecuten más rápido las aplicaciones. El compilador fue construido desde el principio como un conjunto de librerías altamente optimizadas fáciles de extender y de optimizar, y diseñado para las arquitecturas de los chips modernos que existen actualmente. En Xcode 4, todo el compilador de pila del Apple LLVM (*desde el analizador de la parte delantera, hasta el optimizador del código final de vuelta*) tiene un gran soporte para C, Objective-C y C++.

El resultado de la sintaxis, el código de terminación, y cualquier otra característica del índice de activación es manejado por el analizador LLVM. Si el compilador sabe algo acerca de un símbolo, también lo hace el IDE de Xcode. C, C++ y Objective-C todos son exactamente comprendidos en el tiempo de edición, tal y como son en la construcción.

Arreglando y los “Live Themes”:

El motor del Apple LLVM se encuentra trabajando constantemente en segundo plano para entender el código fuente. En el editor, la nueva característica “Live Themes”, utiliza ese conocimiento para que le avise a la codificación de errores a medida que se va escribiendo. Al igual que en un procesador de textos pone de relieve los errores de ortografía, Xcode 4, destaca los errores comunes de codificación, sin necesidad de hacer clic en “Build” en primer lugar.

Solucionar el Problema de forma Automática:

Más allá de los errores de información, el IDE es lo suficientemente inteligente como para solucionar el problema para uno. En muchos casos, el Xcode no sólo informará de un error, también va a presentar una solución. Se tiene que hacer clic en el error para ver que “Fix-its” (*arreglos*) se encuentran disponibles, tales como la corrección de una asignación a una comparación, la reparación de un símbolo mal escrito, o de agregar un punto y coma que haga falta. Con solo un atajo del teclado y al instante se tendrá el error reparado, para que se pueda seguir con la codificación.

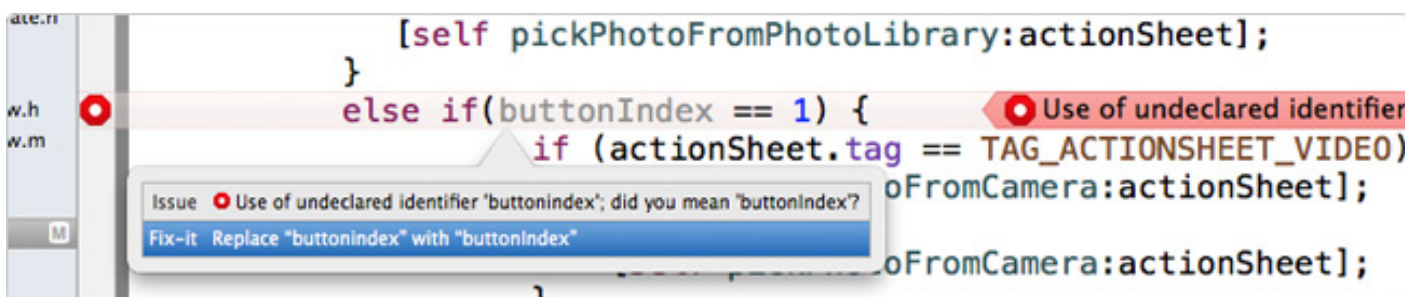


Figura 4.9: Solución de Problemas de forma automática mediante los “Fix-its”.

“Fix-it” es una gran medida que se escribe y que acompaña a las pruebas rigurosas realizadas por la función “Analizar”. El analizador estático de Xcode camina a través de miles de posibles rutas de código, buscando lugares en donde el código fuente, mientras sea válido, se comportan de manera inesperada, como son los errores de asignación de memoria, que nunca golpeó declaraciones de caso, o bucles mal contruidos.

Juntos, “Fix-it” y la Función “Análisis” de Xcode, encuentran las fallas mucho antes que sus usuarios.

Editor de la Versión:

El nuevo Editor de la Versión de Xcode 4 hace que sea fácil de ver cualquiera de las dos versiones del código fuente, de lado a lado, en un editor en vivo. Más importante aún, el Editor de la Versión es una nueva manera de pensar en la gestión de control del código fuente en un IDE, porque la vista de comparación también es una línea de tiempo. Con solo arrastrar el control hacia el centro, y se viaja retrocediendo en el tiempo a través del proyecto, comparando las dos versiones.

El Editor de la Versión también puede mostrar un registro detallado de los acontecimientos pasados, y así realizar un seguimiento por el historial de los “check-ins”. Los Comandos SMC que son complejos, se manejan para uno detrás de las escenas. Inclusive, es posible gestionar múltiples proyectos dentro de un único espacio de trabajo de Xcode 4, un proyecto gestionado en Subversión, Git en el otro, todo esto se actualiza de forma automática.

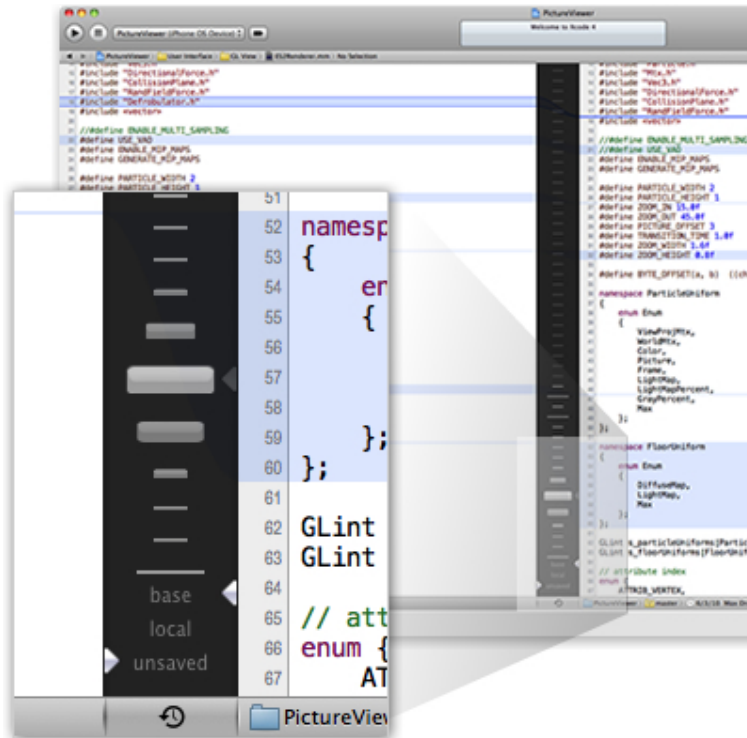


Figura 4.10: Editor de la Versión.

Nuevo Depurador:

Xcode 4 introduce LLDB, un nuevo motor de depuración aportado por Apple, Inc., para el proyecto LLVM.org de código abierto. Al igual que LLVM, el nuevo motor LLDB ha sido diseñado desde cero para consumir menos memoria, y ser un cohete cuando se trata de rendimiento.

El nuevo Motor de Depuración LLDB es el ajuste perfecto para la nueva interfaz de depuración de Xcode 4. Cuando la aplicación se está ejecutando, el navegador mostrará un seguimiento de la pila que se puede expandir o comprimir para mostrar u ocultar los marcos de pila mientras se depura. A medida que se va recorriendo, incluso se puede bloquear a un solo hilo, y luego se hace clic en “Continue”, y seguir el hilo específico de ejecución. La depuración Multicore de Xcode 4 es ahora más fácil, así como también la codificación multi-núcleo con bloques y el “Gran Central Dispatch”.

“Instruments” de Xcode 4:

Instruments tiene una interfaz optimizada, con la nueva Barra de Salto y compresión de la pila de la interfaz de usuario de Xcode 4. Utilizando los nuevos planes de lanzamiento de Xcode 4, es mucho más fácil que antes crear una prueba robusta para ejecutar una aplicación en Instruments. Eso incluye el lanzamiento de Instruments en “modo diferido”, manteniendo la interfaz de usuario de la recopilación de datos fuera de la pantalla y el ahorro de los recursos del sistema para ejecutar y medir la aplicación.



Figura 4.11: Instruments de Xcode 4.

Los nuevos instrumentos de recolección de datos también se encuentran disponibles, incluyendo el OpenGL ES para el seguimiento del rendimiento de los gráficos del iPhone, el seguimiento de la nueva asignación de memoria, que se puede encontrar para el crecimiento de la memoria involuntaria, el “Time Profiler” para el iOS para la recolección de muestras con muy poco gasto, y el completo “System Trace” para conocer a fondo de cómo todos los procesos del sistema interactúan entre sí.

Instruments cubre un área muy grande, ya que tiene una sobrecarga muy baja, y esto lo hace más útil que nunca.

4.2 Desarrollo de Aplicaciones para el iOS:

El iOS es el Sistema Operativo móvil más avanzado del mundo, el cual redefine constantemente lo que la gente puede hacer con un dispositivo móvil. En conjunto, el SDK de iOS y el IDE de Xcode hacen que sea más fácil para los desarrolladores que crean aplicaciones móviles revolucionarias.



Figura 4.12: Desarrollo de Aplicaciones para el iPhone y el iPad.

Fundación Potente:

Derivado del núcleo de las tecnologías OS X, la increíble experiencia del usuario del iOS, ha sido optimizado para aprovechar al máximo el hardware del iPhone, el iPad y el iPod Touch. Estas tecnologías compartidas entre el iOS y el OS X, incluyen el Kernel OS X, sockets de red BSD y los compiladores de Objective-C y C/C++ para un rendimiento nativo.

4.3 iOS Cocoa Touch:



Figura 4.13: Cocoa Touch para el iOS.

Los frameworks de Cocoa Touch para el iOS que impulsan compartir las aplicaciones probadas con varios patrones que se encuentran en la Mac, pero que fueron construidos con un enfoque especial en el tacto y las interfaces basadas en optimización. El UIKit proporciona las herramientas básicas que se requieren para implementar gráficos, basados en eventos de las aplicaciones en iOS. El UIKit se basa en la misma infraestructura del framework “Foundation” que se localiza en el Mac OS X, incluyendo el manejo de archivos, redes y relaciones de cadena, y mucho más.

Desarrollo en Objective-C:

Gran parte de Cocoa Touch se implementa en Objective-C, el cual es un Lenguaje Orientado a Objetos, que se compila para correr a una velocidad increíble, pero emplea un tiempo de ejecución por lo que es verdaderamente dinámico excepcionalmente flexible. Debido a que Objective-C es un superconjunto de C, que es fácil de mezclar C y C++, incluso en las aplicaciones de Cocoa Touch.

A medida que la aplicación se ejecuta, el tiempo de ejecución de Objective-C crea instancias de objetos en función de la ejecución de la lógica, no sólo en formas definidas durante la compilación. Por ejemplo, una aplicación en Objective-C cuando se está ejecutando puede cargar una interfaz (*un archivo .nib creado por Interface Builder*), conectando los objetos de Cocoa en la interfaz con el código fuente de la aplicación, a continuación, se tiene que ejecutar el método correcto una vez que es presionado el botón de la interfaz de usuario. No es necesario volver a compilar.

Variedad Completa de Frameworks:

Además de UIKit, la colección de frameworks de Cocoa Touch incluye todo lo necesario para crear aplicaciones iOS de clase mundial, a partir de gráficos en 3D, al audio profesional, la creación de redes y las API de acceso al dispositivo especial para controlar incluso la cámara, o bien, obtener la ubicación del hardware GPS. Cocoa Touch incluye potentes frameworks de Objective-C que realizan tareas enteras en sólo unas pocas líneas de código fuente la API fundamental del Lenguaje C, que permite el acceso directo al sistema cuando lo requiera.

Ejemplos de estos frameworks se explican a continuación:

- **Core Animation:**

Utilizando Core Animation se crean experiencias de usuario ricas desde el modelo de programación sencillo basado en capas independientes de composición gráfica.

- **Core Audio:**

Core Audio es una tecnología de nivel profesional para la reproducción procesamiento y grabación de audio, por lo que es fácil de añadir características poderosas de audio para la aplicación.

- **Core Data:**

Core Data ofrece una solución orientada a objetos de administración de datos que es fácil de usar y entender, sin embargo, está construido para manejar el modelo de necesidades de datos de cualquier aplicación, sea grande o pequeña.

4.4 Gestión de Datos (iOS):

iOS cuenta con una amplia colección de herramientas y frameworks para almacenar, acceder y compartir datos. El Core Data es un framework que tiene todas las funciones de modelado de datos para las aplicaciones orientadas a objetos Cocoa Touch, mientras que SQLite es perfecto para trabajo de bajo nivel de base de datos relacional. Las aplicaciones pueden compartir los datos mediante un sistema basado en URL que funciona en todos los iOS. En las aplicaciones Web se pueden guardar los datos en la memoria caché del lado del cliente con el API de almacenamiento de datos de HTML5. Las aplicaciones de iOS tienen acceso a los datos globales de un dispositivo, como los Contactos de la Agenda y las Fotos de la Biblioteca de Fotos.

Core Data:



Figura 4.14: Core Data.

Core Data proporciona un framework de modelo de datos flexible y potente para crear aplicaciones Cocoa bien factorizados, basado en el Modelo-Vista-Controlador (*MVC*). Core Data proporciona una solución de uso general de gestión de datos desarrollado para manejar las necesidades del modelo de datos de cada tipo de aplicación, ya sea grande o pequeña. Se puede construir cualquier cosa, desde una aplicación de gestión de contactos hasta un programa de ilustración arte-vectorial en la parte superior de la misma. El límite es el cielo.

Core Data permite definir rápidamente el modelo de datos de la aplicación de forma gráfica y de fácil acceso desde el código fuente. Se proporciona una infraestructura para hacer frente a las funcionalidades comunes como son guardar, restaurar, deshacer y rehacer, lo que le permite seguir adelante con la tarea de construir la innovación en la aplicación. Porque Core Data utiliza el “built-in” de la biblioteca de datos de SQLite, ya que gracias a eso no existe la necesidad de instalar un sistema de base de datos independiente.

Interface Builder, el editor de interfaz de usuario gráfico de Apple, ofrece los objetos del controlador pre-construidos de Core Data, que eliminan la necesidad de que la mayor parte del código de unión entre la interfaz de usuario de la aplicación y el modelo de datos. Para que uno no se tenga que preocupar por la sintaxis de SQL, así como también uno no tenga que mantener los árboles lógicos para rastrear el comportamiento del usuario, y por último, para que uno no tenga que crear un mecanismo nuevo de persistencia. Todo está hecho para uno, cuando uno conecta la interfaz del usuario de la aplicación con el modelo Core Data.

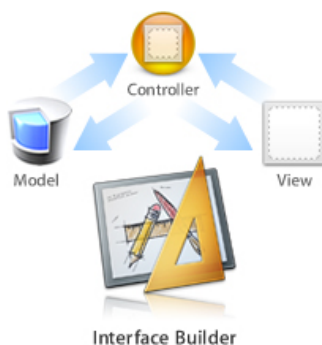


Figura 4.15: Interface Builder.

SQLite:

iOS incluye la popular biblioteca SQLite, un motor de base de datos relacional ligero pero potente que se integra fácilmente en una aplicación. Se utiliza en infinidad de aplicaciones a través de múltiples plataformas, SQLite es considerado un estándar de facto para la programación de base de datos ligera incorporada de SQL. A diferencia del framework orientado a objetos de Core Data, SQLite utiliza un procedimiento, una API de SQL orientada a manipular directamente las tablas de datos.

Compartir Datos entre las Aplicaciones:

iOS ofrece opciones de conectividad potentes para el intercambio de información entre las aplicaciones instaladas en un dispositivo. Usando una sintaxis basada en URL, se puede acceder a los datos de la Web, así como la información pase a otras aplicaciones instaladas, como Mail, iTunes y YouTube. La propia aplicación puede declarar un esquema único de URL, que permite a cualquier aplicación colaborar y compartir datos con la aplicación.

Acceso a Contactos y Fotos:

Los datos y medios de comunicación del iPhone están disponibles para la aplicación a través de APIs seguros y que son fáciles de usar. La aplicación puede crear nuevos Contactos de la Agenda, así como también obtener información de los Contactos ya existentes. Del mismo modo, la aplicación puede cargar, mostrar y editar fotografías desde la Biblioteca de Fotos, así como usar la cámara incorporada para tomar nuevas fotos.

Acceso al Calendario:

El Kit de Eventos permite a las aplicaciones de iOS acceder a la información de eventos de la base de datos del Calendario de un usuario. Los Eventos "Fetch" que están basados en un rango de fechas o un identificador único, recibe las notificaciones de cuando hay cambios de los eventos de los registros, y permite a los usuarios crear y editar eventos de cualquiera de sus Calendarios. Los cambios realizados a los acontecimientos en la base de datos del Calendario de un usuario con el Kit de Eventos se sincronizan automáticamente con el Calendario correspondiente, incluyendo los Calendarios alojados en los Servidores CalDAV y Valores.

Archivos XML:

Los archivos XML proporcionan un formato ligero estructurado para que la aplicación se pueda leer y escribir. Además de los archivos XML, fácilmente encajan en el sistema de archivos del iOS. Se guardan los ajustes de la aplicación y las preferencias del usuario en el "built-in" predeterminado del usuario de base de datos. Este almacén de datos basados en XML, incluye una API simple con características de gran alcance, incluyendo la capacidad para serializar y restaurar objetos complejos bajo demanda.

Almacenamiento del Lado del Cliente HTML5 en Safari:

El Navegador avanzado Safari incluido en el iOS, soporta las últimas funciones de almacenamiento de datos de HTML5 desconectadas. El almacenamiento fuera de línea significa que una aplicación Web puede almacenar los datos de sesión localmente en la memoria caché del dispositivo iPhone o el iPod Touch usando una simple API llave/valor de los datos, o una interfaz más avanzada de SQL. El dato es persistente entre los lanzamientos de Safari, lo que significa poner en marcha aplicaciones más rápido, son menos dependientes de la red, y se desempeñan mejor que nunca.

4.5 Desarrollo para OS X (El Sistema Operativo más avanzado del mundo):

OS X es el Sistema Operativo más avanzado del mundo, el cual está basado en una base probada de UNIX, junto con un equipo de escritorio con aceleración GPU, los servicios de redes intuitivas y todo el sistema de optimización para CPUs multi-núcleo. Y con el entorno de desarrollo Xcode integrado, tiene un potente conjunto de herramientas para crear aplicaciones atractivas, innovadoras y de alto rendimiento.



Figura 4.16: Desarrollo de Aplicaciones para iMac y MacBook (OS X).

Cocoa Frameworks:

Cocoa es un productivo entorno de programación de muy alto nivel que le da acceso completo a toda la potencia del Sistema Operativo, incluyendo la mirada y sentir de la firma Mac. Escrito en el dinámico Lenguaje de Programación Objective-C, Cocoa ofrece experiencias de usuarios que son interesantes y ricos en animación, pero que es fácil de aumentar sus comportamientos personalizados. Con el soporte integrado para el Patrón "Controlador de Vista Modal", Cocoa proporciona una base sólida para la construcción del arte de las aplicaciones. Con la herramienta de Interface Builder, que es a la vez fácil y divertido de arrastrar y soltar los Controles de Cocoa, para diseñar la interfaz de la aplicación del usuario.

Acceso Completo al Sistema Operativo:

OS X integra un número de tecnologías que son fáciles de utilizar, las cuales juegan un doble papel como grandes aplicaciones y servicios del sistema. Siguiendo algunas pautas simples, la aplicación puede trabajar con esos servicios para proporcionar características de gran alcance que pueden ser utilizados por otras aplicaciones. La aplicación puede ser conducido en manos libres con “Automator” o el “AppleScript”, volverse accesible para los usuarios del extranjero a través de “Voice Over”, y buscar rápidamente los documentos mediante “Spotlight”. Cocoa también puede ayudar a integrar la aplicación en estrecha colaboración con el resto del sistema, el registro instantáneo de los documentos a través de “Time Machine”, mostrar el contenido a través de “iChat Theater”, y compartir una base de datos común de la Agenda.



Figura 4.17: Tecnologías de OS X.

Desarrollo de Bellas Aplicaciones:

OS X está construido alrededor de una pila potente e integrada de tecnologías de gráficos, incluyendo Core Animation, Core Image y Open GL. Trabajando juntos, estos frameworks de gráficos hacen posibles características de vanguardia, incluyendo la interfaz de usuario de dibujo en 2D y en 3D, animación y multimedia.

Fundación Potente:



Figura 4.18: Desarrollo de Aplicaciones de 64 bits.

OS X proporciona una excelente estabilidad y rendimiento. Diseñado para las aplicaciones de 64 bits y procesadores multicore. OS X le da una potencia seria. El nuevo “Grand Central Dispatch” asegura que la aplicación hace el mejor uso posible de todo el poder de cómputo disponible, ganando velocidad, sin la complejidad de la programación en hilo. Desarrollando para 64 bits, se aprovecha de un conjunto de instrucciones de la CPU optimizado y con más registros. Terabytes de espacio de dirección, y la posibilidad de utilizar gran cantidad de memoria RAM física. Si se escriben juegos, la secuencia de genes, imágenes médicas avanzadas, o aplicaciones geoespaciales, OS X tiene la estabilidad y el rendimiento para hacer volar la aplicación.

4.6. Mac OS X Cocoa:

Los frameworks de Cocoa y Cocoa Touch que rigen al Mac OS X y el iOS, se encuentran estrechamente integradas en la experiencia de desarrollo Xcode. El API de alto nivel de Cocoa, hacen que sea fácil agregar animaciones, creación de redes y la aparición de la plataforma nativa, y el comportamiento de la aplicación con sólo unas pocas líneas de código fuente.

Cocoa Frameworks:

Los frameworks de Cocoa consisten en bibliotecas, APIs, y tiempos de ejecución que forman la capa de desarrollo para todos los Mac OS X. Al desarrollar con Cocoa, se van a crear aplicaciones de la misma manera que el propio Mac OS X es creado. La aplicación heredará automáticamente los comportamientos grandes y aspecto de Mac OS X, con pleno acceso a la energía subyacente del Sistema Operativo UNIX. El uso de Cocoa con el IDE de Xcode es simplemente la mejor manera de crear aplicaciones nativas de Mac.



Figura 4.19: Cocoa Framework.

El Poder de Objective-C:

Gran parte de Cocoa está implementado en Objective-C, un Lenguaje Orientado a Objetos que se compila para correr a una velocidad increíble, pero emplea un tiempo de ejecución por lo que es verdaderamente excepcionalmente flexible. Debido a que Objective-C es un superconjunto de C, que es fácil de mezclar C y C++, incluso en las aplicaciones de Cocoa.

A medida que la aplicación se ejecuta, el tiempo de ejecución de Objective-C crea instancias de objetos en función de la ejecución de la lógica, no sólo en formas definidas durante la compilación. Por ejemplo, una aplicación de Objective-C en funcionamiento puede cargar una interfaz gráfica (*un archivo ".nib" creado por Interface Builder*), conectando los objetos de Cocoa en la interfaz gráfica con el código fuente de la aplicación, a continuación, se ejecuta el método correcto una vez que es presionado el botón de la interfaz de usuario. No es necesario volver a compilar.

El tiempo de ejecución dinámico de Objective-C es similar al de muchos lenguajes de programación modernos, por lo que es fácil de trazar las características de Cocoa a otros idiomas utilizando el "Cocoa Bridge". Con el "Cocoa Bridge", los desarrolladores pueden crear de primera clase, aplicaciones para Mac OS X con AppleScript, Ruby y Python.

Cocoa utiliza el Patrón de Diseño "Modelo – Vista - Controlador":

Cocoa utiliza el Patrón de Diseño "Modelo-Vista-Controlador" (MVC) en todo. Los Modelos encapsulan los datos de la aplicación, las Vistas muestran y editan los datos, y los Controladores median la lógica de ambos. Al separar las responsabilidades de esta manera, uno termina con la aplicación que es más fácil de diseñar, implementar y mantener.

El Patrón MVC significa que Interface Builder no requiere ningún código que se escribe o se generan mientras uno se centra exclusivamente en la Vista de la aplicación. Los enlaces de Cocoa en la Mac eliminan la mayor parte del código de unión, por lo que la conexión entre los Controladores codificados en Xcode y las Vistas diseñadas en Interface Builder ambos son una simple cuestión de forma gráfica "cableada". Interface Builder funciona con Cocoa para localizar fácilmente la aplicación, para que así uno pueda encontrarlos en mercados completamente nuevos.



Figura 4.20: Patrón de Diseño *Modelo-Vista-Controlador*.

4.7 Proceso de Desarrollo de una Aplicación para el iPhone y el iPad:

El Proceso de Desarrollo de una Aplicación para el iPhone, iPod Touch y el iPad, pasa por un Ciclo de Vida, además de conocer la Plataforma y las Tecnologías a utilizar, que a continuación se presenta un Resumen.

Primero, hay que tener en cuenta que desarrollar para un dispositivo móvil no es lo mismo que para una computadora, es decir, no es lo mismo hacer una Aplicación para el iPhone que para una Mac. Los Motivos de tal Afirmación son los siguientes:

Tamaño de la Pantalla:

El Tamaño de la Pantalla es considerablemente menor que en una Mac, por tanto el desarrollador debe tener en cuenta dicho aspecto durante todo el desarrollo y más concretamente durante el Diseño de la Interfaz de Usuario.

El tamaño de la pantalla del iPhone es de 480x320 pixeles y del iPad es de 1024x768 pixeles, por lo tanto la Interfaz de Usuario se debe centrar en lo esencial que el usuario pueda necesitar en cada pantalla, o mejor dicho, en cada Vista.

El Uso de la Memoria:

El uso de la memoria es crucial en cualquier software, pero cuando se habla de dispositivos móviles aún más. El iPhone no dispone de espacio de Cache para la Memoria, luego hay que tener especial cuidado con el uso de la memoria que la aplicación lleva a cabo en cada momento. Para ello, se ofrece al desarrollador la Aplicación "Instruments", a través de la cual se puede monitorear el uso que está realizando nuestra aplicación en cada momento.

El iPhone, así como también el iPod Touch y el iPad introducen un mecanismo a través del cual se produce un evento cuando el dispositivo se está quedando sin memoria. En el caso de que no se haga nada, la aplicación se cerrará inmediatamente, por tanto es nuestro deber (por el bien del usuario), implementar que hacer en caso de que se produzca dicho evento o aviso (guardar los datos, estado de la aplicación, etc.).

Aplicaciones Simultáneas:

El iPhone sólo permite que haya una Aplicación corriendo o ejecutándose (Aplicaciones desarrolladas por Terceros, este es el motivo por el cual no es posible introducir una Máquina Virtual de Java y desarrollar Aplicaciones en Java, por el momento) a la vez.

Luego si se está utilizando nuestra aplicación y el teléfono suena o llega un mensaje SMS, la aplicación se cerrará (entra en Modo Pausa) y es nuestro deber guardar tanto los datos como el estado de la aplicación lo más rápido posible, para ofrecer al usuario un entorno rico y agradable, y no ofrecerle una experiencia de usuario negativa.

Ventanas Simultáneas:

Esta es una de las grandes diferencias entre una Aplicación de Escritorio y una Aplicación para Dispositivo Móvil, y es que el iPhone sólo permite una ventana simultánea en el Dispositivo. Dicha ventana puede tener varias vistas, pero el Usuario sólo accederá a ellas de manera secuencial y nunca simultáneamente.

Simplicidad Absoluta:

Este es uno de los aspectos en los que actualmente muchas de las Aplicaciones para Dispositivo Móvil carecen. Y es que el Usuario una vez arrancada la Aplicación nunca debe preguntarse que hacer o como hacer tal cosa. La Simplicidad de la Aplicación debe ser Absoluta evitando en primer lugar tener que recurrir a implementar una ayuda.

Tipos de Aplicaciones:

Una vez conocido los aspectos anteriores, hay que decidir qué Tipo de Aplicación desarrollar. Existen Tres Opciones actualmente:

1. Aplicación Web:

Aplicación basada solamente en Contenido Web, adaptado a la pantalla y uso del iPhone.

2. Aplicación Nativa:

Aplicación que reside en el Teléfono y que hace uso de todas las Tecnologías del iPhone. Es nuestro propósito para la Aplicación de ejemplo desarrollada.

3. Híbrido entre los dos Anteriores:

Aplicación nativa en el iPhone, pero que parte de sus Funcionalidades e infraestructura la lleva a cabo a través de una Aplicación Web. Aunque no es el cometido de nuestra Aplicación de ejemplo, si es un buen punto de partida y por donde seguro irán muchos Desarrolladores de Aplicaciones para el iPhone.

Estilo de las Aplicaciones:

Decidido qué Tipo de Aplicación se va a desarrollar, hay que saber que existen, fundamentalmente, tres Tipos de Estilo de Aplicaciones que son:

1. Aplicaciones Productivas:



Figura 4.21: Aplicaciones Productivas.

Las Aplicaciones Productivas son esa Clase de Aplicaciones donde resalta ante todo la organización y manipulación de la información. El usuario en este Tipo de Aplicaciones, agradece ante todo encontrar fácilmente la información, sin prestar demasiada atención a la interfaz del usuario.

La información, en este Tipo de Aplicaciones, se aconseja mostrarla de lo más general a lo más específico, permitiendo que sea el usuario quien navega si lo desea de menús más generales a menús con un nivel de detalle mayor.

2. Aplicaciones de Utilidad:



Figura 4.22: Aplicaciones de Utilidad.

Las Aplicaciones de Utilidad se caracterizan por requerir la mínima acción del usuario para mostrar la información. Suelen ser visualmente atractivas y la información se organiza en una sucesión de visitas, en contraste con las productivas que la información se organiza de manera diferente.

El usuario no necesita navegar en vistas más específicas para encontrar la información, sino que el tipo de información es la misma en cada vista. El usuario navega por las diferentes vistas para encontrar la misma información pero de diferentes objetos.

3. Aplicaciones a Pantalla Completa (FULL SCREEN):



Figura 4.23: Aplicaciones a Pantalla Completa (FULL SCREEN).

Este Tipo de Aplicaciones ofrecen todo su contenido a pantalla completa y con una riqueza visual excelente. Videojuegos, Música o Reproductores de Video son las que aprovechan al máximo este recurso en este Tipo de Aplicaciones, donde lo que busca el usuario es divertirse y pasar un buen rato a gusto.

Se basan en el Contenido Visual y en la experiencia de usuario. Utilizan una cantidad enorme de información, pero que no se muestra al usuario, por lo que éste no necesite que se le muestren barras de navegación o métodos de acceder a la información, como sucede en las dos Clases anteriores de Aplicaciones.

Entorno de Ejecución de una Aplicación:

Para conocer el Entorno de Ejecución de las Aplicaciones es fundamental y último requisito que como Desarrolladores debemos estudiar para poder abordar el Desarrollo de la Aplicación de forma eficiente.

El Entorno de Ejecución de una Aplicación en el iPhone, así como en el iPod Touch y el iPad, está continuamente recibiendo Eventos del Sistema y respondiendo a dichos Eventos presentándose al usuario de manera adecuada. Este es el Patrón de Diseño llamado “**Event and Drawing Cycle**”.

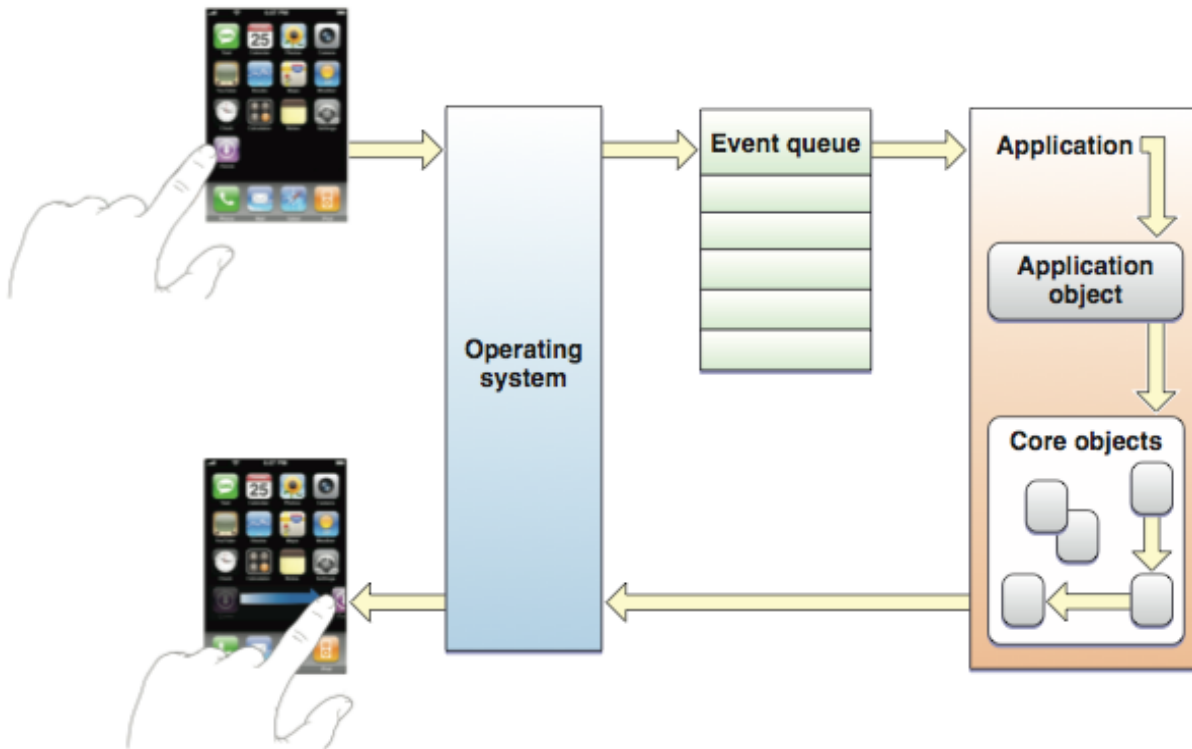


Figura 4.24: Diagrama de Apple Developers que ilustra el *Event and Drawing Cycle*.

Una vez que la Aplicación se ejecuta, el Sistema Operativo del iPhone llama a la Función Main (punto de entrada en cualquier Aplicación). La Clase Principal es la UIApplicationMain, la cual inicializa el “Event and Drawing Cycle”.

```
#import <UIKit/UIKit.h>
int main(int argc, char+argv[])
{
    NSAutoreleasePool+pool = [NSAutoreleasePool new];
    UIApplicationMain(argc, argv, @"UIApplication", @"AppController");
    [pool release];
    Return 0;
}
```

Figura 4.25: *Main* en una Aplicación del iPhone.

El Sistema detecta los Eventos ocurridos en el Dispositivo móvil, y los coloca en “la Cola de Eventos de la Aplicación”. El Objeto de la Aplicación selecciona el Evento Superior de la Cola y lo delega en el Objeto determinado para su Ejecución.

Ciclo de Vida de una Aplicación:

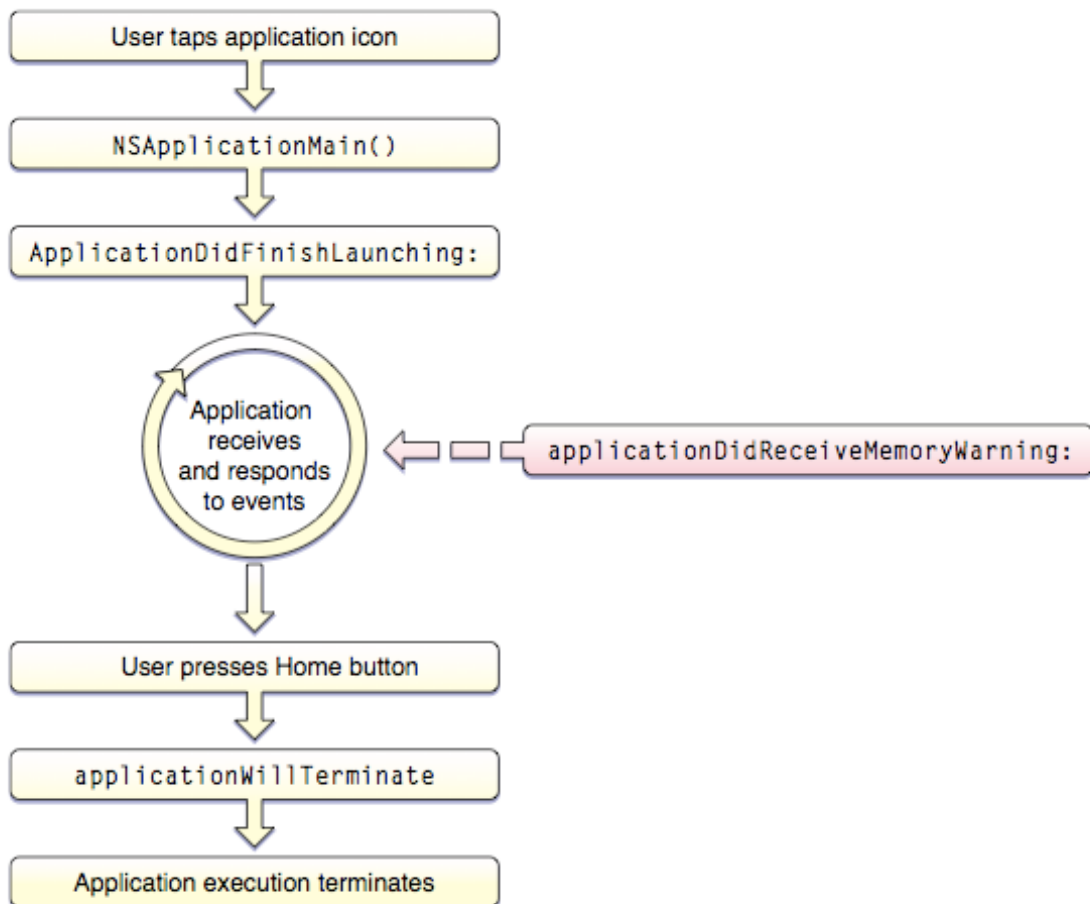


Figura 4.26: Ciclo de Vida de una Aplicación para Dispositivo Móvil.

Como se muestra en el Diagrama anterior, una vez que el usuario presiona el ícono de la Aplicación, se inicia el “Event and Drawing Cycle”, el cual captura los Eventos y responde ante ellos. Pero existen Eventos que pueden parar dicho Ciclo de Ejecución, como puede ser que el usuario presione el botón de Home o reciba una Llamada de Teléfono o recibe una Notificación de Falta de Memoria.

Al sólo poder ejecutarse una Aplicación simultáneamente, la actual se cerrará terminando su ejecución.

4.8 Historia :

Xcode se lanzó el 24 de octubre de 2003, junto con la versión 10.3 de Mac OS X (Panther), siendo desarrollado a partir del anterior entorno de desarrollo de Apple, que se llamó "Project Builder", al que sustituyó; éste programa a su vez, también era una herencia de la compañía NeXT, que se fusionó con Apple en 1996.

La aparición de Xcode 2.1 en junio de 2005 fue significativa, ya que proporcionó a la comunidad de desarrolladores las herramientas para crear binarios universales, que permiten al software creado para Mac OS X ser ejecutado tanto en la arquitectura PowerPC como en la nueva, que estaba basada en Intel (x86). Esta versión integró además, las herramientas y marcos de trabajo "WebObjects" de Apple para construir aplicaciones y servicios web de Java, que antes se vendían como un producto separado.

Con el lanzamiento de Mac OS X versión 10.5, también lo fue Xcode 3.0, que tenía como principales novedades, la inclusión de Objective-C 2.0, un nuevo Interface Builder, además de la opción de refactorizar proyectos y hacer "snapshots" (copias de seguridad) del proyecto entre otras.

Xcode 4, fue lanzado a principios de 2011, incluye novedades como una nueva interfaz y la compatibilidad con Mac OS X 10.8 Mountain Lion. Con esta versión, Xcode dejó de ser compatible con Mac OS X 10.5 Leopard.

CAPITULO No. 5: “ANALISIS Y DISEÑO DEL SISTEMA”

5.1 Análisis del Sistema:

En este Capítulo, se explica con detalle el desarrollo completo del Sistema llamado “**iShelper**”, que es una aplicación que se va a usar en el iPad, su objetivo es el de generar reportes de soporte técnico vía inalámbrica, para que se realice de forma más rápida y eficiente el trabajo realizado por los técnicos que se encuentran en el Área de Soporte Técnico de una Empresa u Organización (ya sea Pública o Privada).

Mediante el desarrollo de la aplicación se desea mostrar una nueva forma de trabajo, el cual consiste en reducir el tiempo que actualmente se hace al momento de generar un reporte de soporte técnico dentro de una Empresa u Organización.

Algunos de los puntos importantes e interesantes que tiene este Sistema, el cual se va a realizar a través de un iPad, se mencionan a continuación:

- Comunicaciones por HTTP.
- Uso de bases de datos.
- Acceso a la información de los Reportes.
- Acceso a la información sobre el mismo dispositivo.
- Composición de interfaces de usuario.
- Gestión de opciones de Menú.
- Control de la conexión Wi-Fi.
- Envío de correos electrónicos.

Introducción:

El objetivo principal de este Sistema es el de generar reportes de soporte técnico, tanto de Mantenimiento Correctivo, así como el de Mantenimiento Preventivo, para que el técnico del Área de Soporte Técnico de la Empresa en la que trabaja, le resulte de una forma más fácil y amigable la elaboración del reporte, para que después lo envíe hacia el Servidor Web vía inalámbrica, que a su vez se aloja en una base de datos, en la cual posteriormente se revisan todos los reportes para darles un seguimiento oportuno, y todo esto se genera desde el iPad.

Soporte Técnico:

Es un rango de servicios que proporcionan asistencia con el hardware y/o software de las computadoras, impresoras e inclusive cualquier otro dispositivo electrónico o mecánico. En general, los servicios de soporte técnico tratan de ayudar a los usuarios a resolver los problemas que tengan con algún equipo, en vez de entrenar o personalizar.

La mayoría de las Empresas que se dedican a vender software y/o hardware, ofrecen el soporte técnico vía telefónica o en línea. Las Organizaciones, Empresas e Instituciones por lo general, cuentan con sus propios técnicos que laboran dentro del Área de Soporte Técnico.

El Soporte Técnico y las Tecnologías de la Información y las Comunicaciones (TICS):

En el mundo empresarial, el soporte técnico se realiza de manera integral, a través de algún medio de contacto con los usuarios (por ejemplo: teléfono, correo electrónico, aplicación en línea, etc.), recibe el nombre de "Help Desk". Cuando un usuario llama al "Help Desk" para reportar alguna anomalía en el funcionamiento de las TICS que se encuentran a su cargo, se procede a registrar el requerimiento en un Sistema de Información, se le asigna un responsable para que lo atienda, y para que posteriormente se le dé un seguimiento a su caso.

Como en cualquier otro proceso del mundo empresarial, existen algunos estándares para la buena administración de este tipo de servicios a las TICS, entre las características que tienen, existe una que vale la pena resaltar, la cual tiene la intención de alinear las actividades que tiene en torno a las TICS con las necesidades estratégicas de la Empresa.

Entre los Estándares más reconocidos actualmente son los siguientes:

- ITIL (Information Technology Infrastructure Library).
- COBIT (The Control Objectives for Information and related Technology).
- Norma ISO 20000 (Gestión de Servicios de Tecnologías de la Información).

Aunque estos Estándares son ampliamente utilizados por las grandes Empresas y llenar los requisitos que demandan es costoso, las Instituciones Educativas (IE) deben conocerlos y tratar de utilizar algunos de sus parámetros, no solo para controlar la calidad del servicio que reciban, si no además, para agilizar y mejorar el soporte técnico que dentro de la Institución se ofrece tanto al área administrativa, así como también al área académica.

Por lo regular, en las Instituciones Educativas de Nivel Básico y Medio Superior, el soporte técnico suele estar a cargo de el docente de Informática/Sistemas/TICS, un técnico externo o una empresa pequeña de soporte técnico, que cuente con técnicos que se dedican a realizar el Mantenimiento que requieran los equipos informáticos, periféricos, redes, software, etc.

Cuando el soporte técnico lo ofrece el docente de Informática, es posible que solo pueda atender algunos casos sencillos en cuanto al Mantenimiento Correctivo se refiere, es decir, puede solucionar algunos problemas que le reporten y que su solución sea fácil de resolver; otros problemas que son más complejos como es el daño de una pieza o componente de algún equipo informático, por ejemplo el disco duro, la tarjeta madre o problemas con la red (tanto el cableado, así como también su configuración), tienen que ser atendidos por el personal que está capacitado para resolver este tipo de problemas.

De todas formas, no es común que los docentes dispongan del tiempo necesario, o que cuente con la preparación adecuada, para que realice el Mantenimiento Preventivo o Predictivo, pues estos exigen todo un sistema de planeación.

Mantenimiento de Computadoras:

Consiste en actividades físicas y lógicas que buscan reducir la posibilidad de fallas en los equipos de cómputo (incluyendo además de las computadoras, el mouse, impresoras, teclados, monitores, etc.), e inclusive la corrección de fallas menores.

Mantenimiento Preventivo:

Se define como el conjunto de actividades encaminadas a proteger los equipos de cómputo de posibles fallas, utilizando algunos métodos de limpieza física, así como también, métodos basados en el uso del software.

Antes de que inicie el proceso de Mantenimiento Preventivo, se deben tomar algunas recomendaciones como las siguientes:

- Se verifica que el equipo no se encuentre dentro del Periodo de Garantía.
- Se verifica que la Configuración básica del "Setup", antes y después del Servicio de Mantenimiento, para que así se pueda comparar de que sean idénticas.
- Se verifica que el Sistema Operativo funcione correctamente, y que los periféricos que se encuentren conectados a la computadora funcionen bien (por ejemplo, el mouse, las unidades ópticas, lector de CD y DVD, etc.).

- Realizar un respaldo de los datos del usuario (Backup), si es posible, de todo el contenido del Disco Duro, sino, solo de la información más importante.
- Es necesario realizar un inventario físico del equipo en presencia de un responsable, que en este caso es uno de los técnicos del Área de Soporte Técnico, para que dé fe de las condiciones físicas y de velocidad en que se encuentran los equipos informáticos (velocidad del procesador, cantidad de memoria RAM, la capacidad del disco duro o la unidad SSD, el número de unidades de almacenamiento con que cuenta, etc.).

En general, el Mantenimiento se tiene que dar a todos los accesorios que están conectados al Gabinete, así como los elementos del interior de éste y el Software:

- Interiores: Tarjeta principal, disco duro, memoria RAM, fuente de alimentación, procesador, tarjetas de expansión, ventilador, unidad de CD, unidad de DVD, etc.
- Exteriores: Pantalla LCD/Monitor CRT, teclado, cámara web, bocinas, escáner, impresoras, etc.
- Software: Actualizaciones del Sistema Operativo, sistema de archivos, programas residentes en memoria, antivirus, memoria virtual, etc.

Mantenimiento Correctivo:

Este tipo de Mantenimiento se aplica para todos los componentes que se encuentran conectados a la computadora, así como a la misma computadora, más el cambio de partes que no necesariamente funciona de manera correcta, que tienen un cierto nivel de fallas que no requieran una reparación como se mencionan a continuación:

- Limpieza de tarjetas con latas de limpiador de circuitos.
- Limpieza de unidades de CD, DVD y digitales.
- Limpieza de conectores, puertos, ranuras de expansión, pines de procesador, etc., utilizando alcohol isopropílico.
- Se utilizan latas de aire comprimido, aspiradora o una brocha en el interior de la computadora, así como también de otros dispositivos conectados a ella.
- En el caso del teclado, se desmontan todas las teclas, y se limpia el interior del mismo.

- Limpieza de virus y actualización del antivirus, así como también instalar las actualizaciones del Sistema Operativo, desfragmentación del disco duro (reacomodo de archivos), revisión de errores en el sistema de archivos, optimización de la memoria RAM y la memoria SWAP.
- El cambio de partes que tienen indicios de fallas, pero que no requieren una reparación como tal.

Mantenimiento Predictivo o “Basado en Condiciones”:

Este tipo de Mantenimiento busca que se realice en el momento que realmente se requiere; para lograrlo, es necesario hacer mediciones continuas de algunas variables, o inclusive conocer la vida útil promedio de algunos componentes, a fin de reemplazarlos justo antes de que fallen.

Un ejemplo puede ser el caso del hardware de una Institución Educativa, en el cual se puede realizar el Mantenimiento Predictivo a una impresora láser cuyo tóner se espera que dure cierto número de impresiones; el administrador de la red puede verificar de manera automática el número de impresiones que se han realizado durante cierto tiempo, y así tener listo el cartucho de repuesto en el momento en el que se necesite cambiarlo.

Otro ejemplo, son las bombillas de los proyectores de video, los cuales tienen un determinado número de horas de vida especificadas por el fabricante; en este caso, se tiene que llevar un registro del uso del equipo, el cual permitirá predecir el momento en el que la bombilla va a dejar de funcionar, y que de esta forma podrá presupuestarse para que se compré la bombilla para cambiarla en el momento que se necesite.

¿Cada cuándo hay que dar el Mantenimiento al Equipo Informático?

Para darle un adecuado Mantenimiento al Equipo Informático, hay que seguir las siguientes recomendaciones:

- Cada mes: Para los equipos que se utilizan frecuentemente, esto es, por lo menos ocho horas diarias, durante los cinco días de la semana, como pueden ser las Oficinas de Gobierno, Centros de Captura de Datos, Servicios de Internet públicos, Escuelas, etc.
- Cada seis meses: Para los equipos que permanecen encendidos de manera regular, como los que se utilizan en los hogares.
- Cada año: Para los equipos que se usan poco tiempo.

Actualización del Equipo de Cómputo:

Esto es muy importante, ya que consiste en reciclar los elementos de una computadora que todavía se encuentren en buen estado, para que así se pueda añadirle otros elementos para que aumente su capacidad (por ejemplo: un disco duro de mayor capacidad, mayor cantidad de Memoria RAM, un procesador más veloz, una fuente de alimentación con mayor potencia, etc.).

El Factor Humano y su Importancia (Humanware):

Es un elemento muy importante en la consecución de los equipos de cómputo, por lo que se necesita también de "Mantenimiento", esto se refiere a que va a requerir de un espacio que sea adecuado para que trabaje con el equipo, un salario, el cual le permita motivarse a que cuide su herramienta de trabajo, así como también que goce de buena salud para que tenga un control consciente de su computadora, herramientas de software que sean fáciles de utilizar, entre otros más.

5.2 Diagramas de Flujo:

Mantenimiento Preventivo:

Objetivo:

Proporcionar el Mantenimiento Preventivo a los Equipos Informáticos de la Empresa, con el fin de mantenerlos en buenas condiciones de uso y así poder evitar fallas menores.

Responsable	No.	Descripción
Área de Soporte Técnico	1	Elabora el programa periódico del Mantenimiento Preventivo a los equipos de cómputo.
	2	Informa a la Dirección General de la Empresa, la fecha y hora en que se dará el Mantenimiento a los equipos de cómputo.
	3	Informa al técnico de la Empresa, para que lleve a cabo el servicio de Mantenimiento Preventivo en la fecha fijada.
Técnico	4	Se presenta en el lugar del usuario en la fecha fijada, y realiza el Mantenimiento Preventivo.
	5	Elabora el reporte del Mantenimiento en las oficinas, en el cual se detallan las condiciones en que se encuentra el equipo informático en ese momento.
	6	Recaba la firma del usuario en el reporte del Mantenimiento.
	7	Entrega los reportes de Mantenimiento Preventivo, efectuados durante el día al Director de Soporte Técnico, de forma impresa y electrónica.
Área de Soporte Técnico	8	Si el equipo no se encuentra en buenas condiciones por presentar fallas mayores, se cataloga en el Programa de Mantenimiento Correctivo. El usuario levanta una orden de servicio en la Mesa de Servicio para su inmediata atención.

	9	Recaba la firma del usuario en el reporte, y posteriormente archiva el reporte de Mantenimiento.
--	---	--

Figura 5.1: Tabla con el Procedimiento del Mantenimiento Preventivo.

Diagrama de Actividades:

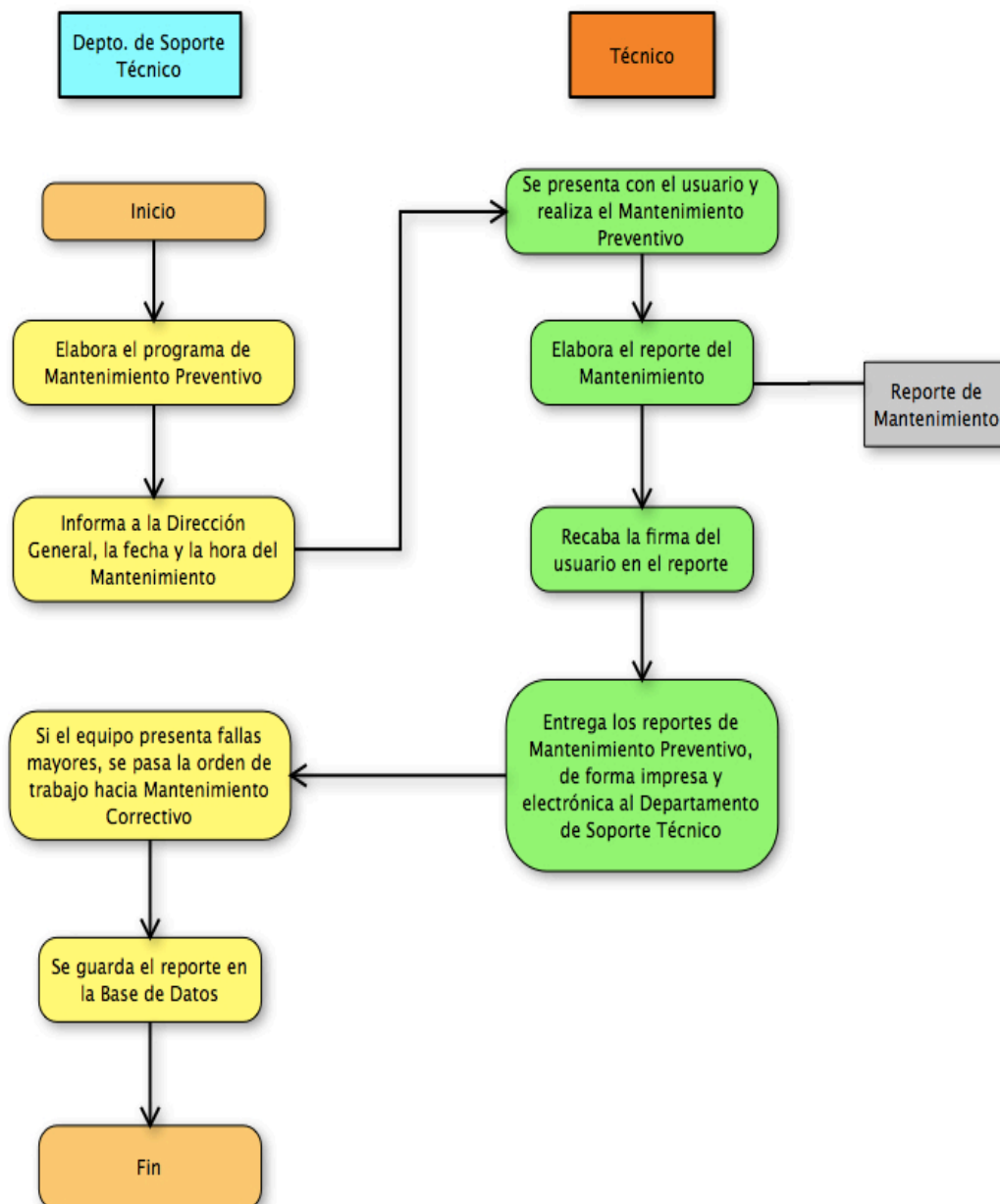


Figura 5.2: Diagrama de Flujo del Procedimiento del Mantenimiento Preventivo.

Mantenimiento Correctivo:

Objetivo:

Proporcionar el Mantenimiento Correctivo a los Equipos Informáticos de la Empresa, con el fin de resolver lo más pronto posible las fallas que se lleguen a presentar, para que los empleados puedan seguir haciendo su trabajo de forma continua.

Responsable	No.	Descripción
Mesa de Servicio	1	Administra y envía el reporte de servicio, que canaliza según corresponda al encargado del sitio.
	2	Recibe el reporte de servicio y realiza la evaluación técnica personalmente, vía telefónica o remota.
Técnico	3	Resuelve el problema.
Usuario	4	(SI) Revisa, acepta y firma de conformidad la reparación del equipo informático.
Técnico	5	Anota las acciones llevadas a cabo en el reporte de servicio, con la solución realizada y envía a la Mesa de Servicio para su captura. Además envía un correo electrónico con el número de reporte terminado.
Mesa de Servicio	6	Recibe el reporte y realiza las actividades tales como su captura electrónica, entre otras más, que se encuentran en el "Manual de Mesa de Servicio".
Técnico	7	(NO) Resuelve el problema.
	8	Anota las acciones llevadas a cabo en su reporte con diagnóstico y lo envía a la Mesa de Servicio, informa si el equipo informático se encuentra en Garantía o es Rentado.
Mesa de Servicio	9	Recibe el reporte y realiza las actividades tales como su captura electrónica, entre otras más, que se encuentran en el "Manual de Mesa de Servicio". VIENE DEL PASO No. 4.- EN CASO DE QUE SI RESUELVE EL PROBLEMA.

Técnico		(NO) Anota las acciones llevadas a cabo en el reporte de servicio, con la solución realizada y envía a la Mesa de Servicio para su captura.
Usuario	10	Acepta y No firma de conformidad.
Mesa de Servicio	11	Administra y canaliza al área correspondiente, si el reporte de servicio es para el área de Garantía.
Área de Garantía	12	Elabora la orden de atención a Garantía, monitorea los avances al Proveedor y a su Gerente; además de que supervisa la reactivación o la reposición del equipo informático. SI APLICA PARA EL PROVEEDOR.
Mesa de Servicio	13	Administra y canaliza al área correspondiente, si el reporte de servicio es para el área de Garantía del Proveedor.
Proveedor	14	Recibe el reporte y endosa su Número de Control Interno de atención a Mesa de Servicio.
	15	Recibe el diagnóstico de Mesa de Servicio y asiste al sitio, atiende y resuelve. Toma nota de las acciones llevadas a cabo en el reporte, y monitorea los avances a Mesa de Servicio.
Usuario	16	Revisa, acepta y firma de conformidad la reparación o cambio del equipo.
Proveedor	17	Endosa el reporte con la solución realizada, incluyendo un correo electrónico dirigido a la Mesa de Servicio titulado "Requerimiento No. Terminado".
	18	Anota las acciones llevadas a cabo en el reporte de servicio, y envía a la Mesa de Servicio, según los lineamientos del Contrato. FIN DEL PROCEDIMIENTO

Figura 5.3: Tabla con el Procedimiento del Mantenimiento Correctivo.

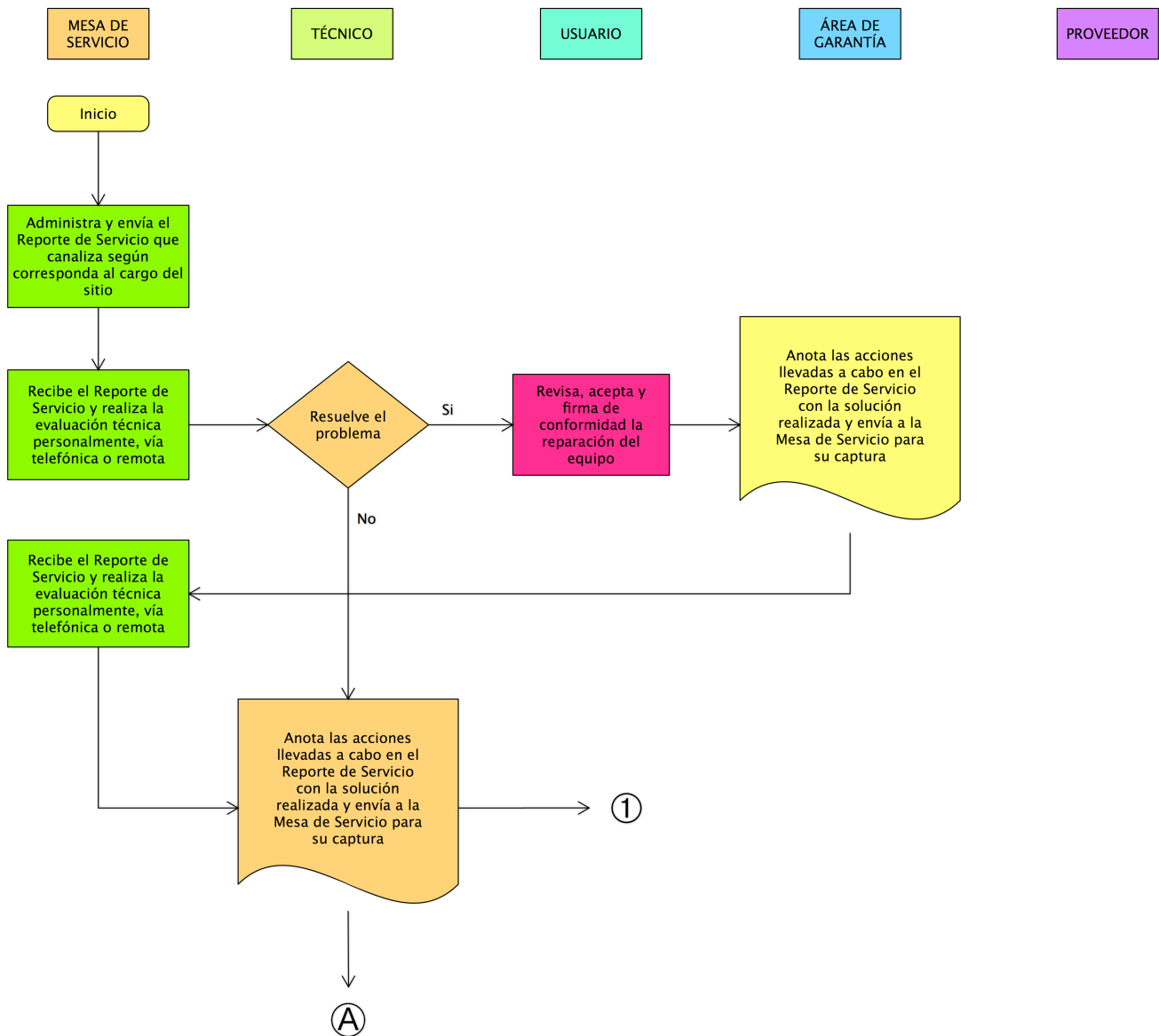


Figura 5.4: Diagrama de Flujo del Procedimiento del Mantenimiento Correctivo (I)

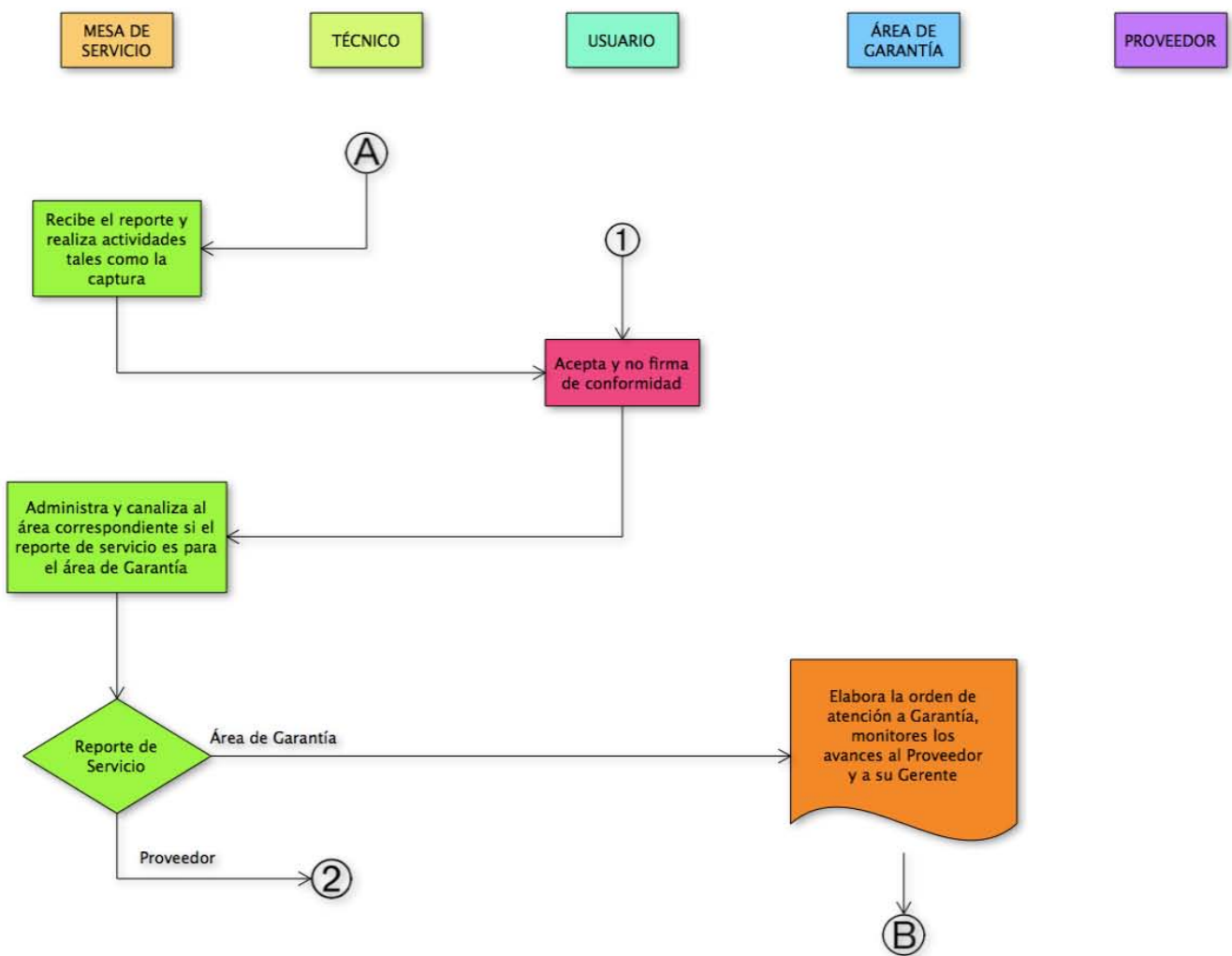


Figura 5.5: Diagrama de Flujo del Procedimiento del Mantenimiento Correctivo (II)

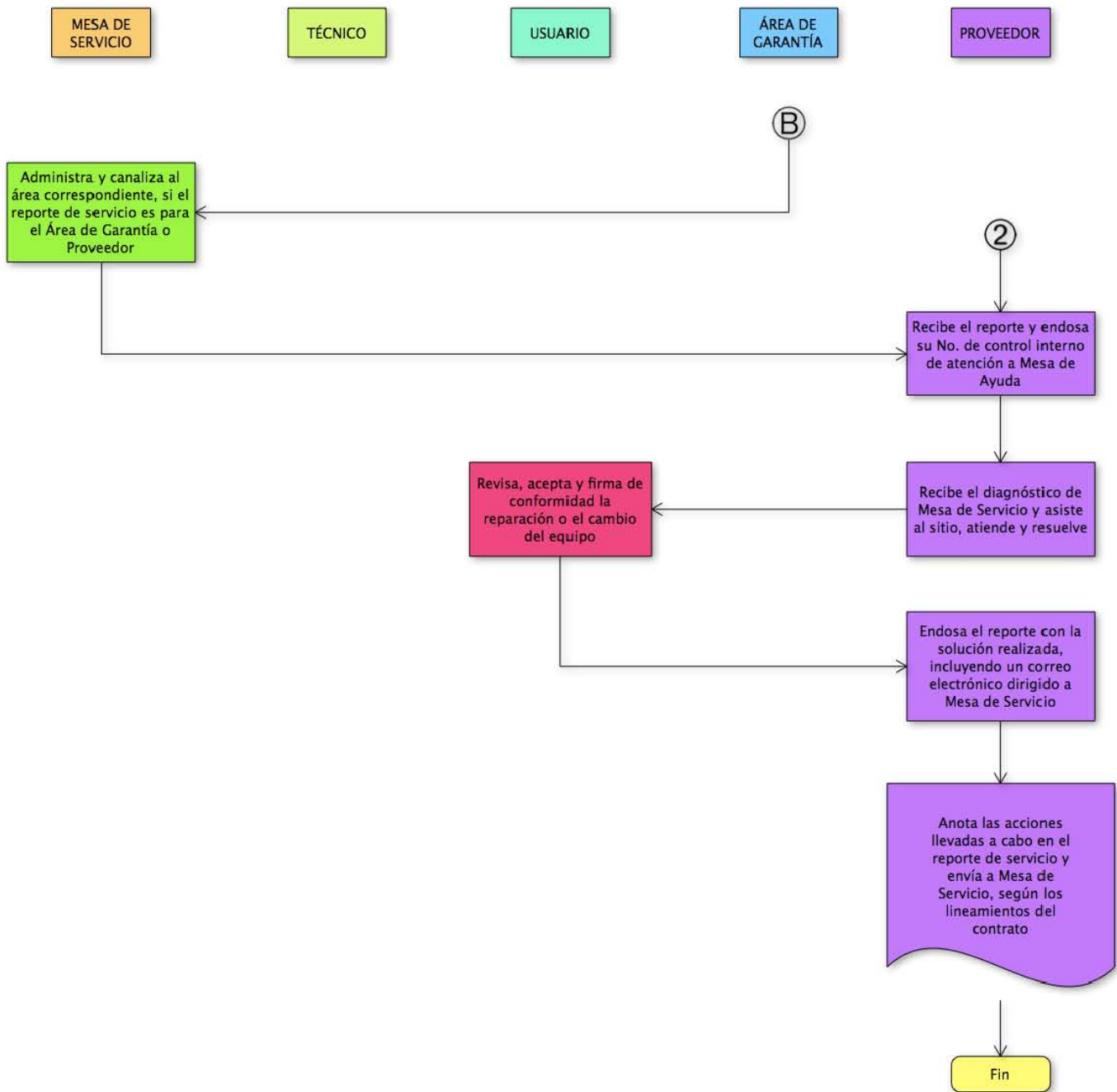


Figura 5.6: Diagrama de Flujo del Procedimiento del Mantenimiento Correctivo (III)

5.3 Diseño de la Base de Datos:

Tablas:

La base de datos guardará únicamente la información que es útil para el Sistema y su correcto funcionamiento, por lo que analizando el objetivo del Sistema, sus aplicaciones y el contenido de lo que se desea obtener en la pantalla de la aplicación, se diseñaron cuatro Tablas para la base de datos del Sistema, las cuales están normalizadas y relacionadas para obtener un correcto funcionamiento del Sistema.

A continuación, se describen las Tablas que contendrá la base de datos.

Nombre: Cat_Empleado.

Objetivo: Esta Tabla contiene la información en una lista de cada uno de los empleados y al Departamento que corresponden, que laboran en una Empresa.

Nombre Campo	Tipo	Longitud	Valores Nulos	Llave
Clave_Empleado	varchar	10	Not Null	Primary Key
Nombre_Empleado	varchar	50	Null	No
Departamento	varchar	20	Null	No

Figura 5.7: Tabla con los Datos para la Base de Datos de la Clase *Empleado*.

Nombre: Cat_Equipo.

Objetivo: Esta Tabla contiene la información de los equipos informáticos que hay en la Empresa.

Nombre Campo	Tipo	Longitud	Valores Nulos	Llave
Clave_Equipo	varchar	10	Not Null	Primary Key
Equipo	varchar	20	Null	No
Marca	varchar	20	Null	No
Modelo	varchar	20	Null	No
Num_Serie	varchar	30	Null	No

Figura 5.8: Tabla con los Datos para la Base de Datos de la Clase *Equipo*.

Nombre: Cat_Reporte.

Objetivo: Esta Tabla contiene la información generada de cada uno de los reportes recabados por el técnico.

Nombre Campo	Tipo	Longitud	Valores Nulos	Llave
Clave_Reporte	varchar	10	Not Null	Primary Key
Tipo_de_Reporte	varchar	15	Null	No
Fecha	varchar	15	Null	No
Hora_Inicio	varchar	15	Null	No
Hora_Final	varchar	15	Null	No
Falla_Reportada	varchar	300	Null	No
Falla_Encontrada	varchar	300	Null	No
Observaciones	varchar	400	Null	No

Figura 5.9: Tabla con los Datos para la Base de Datos de la Clase Reporte.

Nombre: Cat_Técnico.

Objetivo: Esta Tabla contiene la información del técnico que ingresará al Sistema, además de que es una lista de todos los técnicos que tienen acceso al Sistema.

Nombre Campo	Tipo	Longitud	Valores Nulos	Llave
Clave_Técnico	varchar	10	Not Null	Primary Key
Nombre_Técnico	varchar	50	Null	No

Figura 5.10: Tabla con los Datos para la Base de Datos de la Clase Técnico.

5.4 Diagramas de Clases:

En el Sistema se tendrá la información muy clara acerca de cuáles van a ser los datos que se requieren, así como también los que se van a mostrar en la pantalla y guardar en la base de datos, ya que este Sistema va a manejar una información moderada a comparación de otros Sistemas, por lo que se ha concluido que la mejor forma de optimizar el manejo de la información almacenada y buscando la optimización de los recursos, es creando objetos específicos para este Sistema, con la que se podrá almacenar la información que se requiera, creando así un espacio de memoria específico para cada uno de los datos que se tengan, reduciendo de tal forma la pérdida de espacio en memoria y lentitud en el Sistema.

Se crearon cuatro clases distintas para poder manipular la información dentro del Sistema, las cuales se mencionan a continuación:

Empleado:

Objetivo: El objetivo de este clase es crear espacio de memoria específico para almacenar un empleado con todos los datos que se le encuentren en la base de datos.

EMPLEADO
Clave_Empleado
Nombre_Empleado
Departamento

Figura 5.11: Diagrama de Clases de la Clase *Empleado*.

Equipo:

Objetivo: El objetivo de este clase es crear espacio de memoria específico para almacenar la información específica de cada equipo informático que el técnico desee consultar por medio del Sistema.

EQUIPO
Clave_Equipo
Equipo
Marca
Modelo
Num_Serie

Figura 5.12: Diagrama de Clases de la Clase *Equipo*.

Reporte:

Objetivo: El objetivo de este clase es crear espacio de memoria específico para almacenar un reporte que realiza uno de los técnicos sobre alguno de los equipos informáticos de la Empresa, según si es de Mantenimiento Preventivo o de Mantenimiento Correctivo.

REPORTE
Clave_Reporte
Tipo_de_Reporte
Fecha
Hora_de_Inicio
Hora_Final
Falla_Reportada
Falla_Encontrada
Observaciones

Figura 5.13: Diagrama de Clases de la Clase *Reporte*.

Técnico:

Objetivo: El objetivo de este clase es crear espacio de memoria específico para almacenar la información básica de un técnico existente en la base de datos del Sistema.

TÉCNICO
Clave_Técnico
Nombre_Técnico

Figura 5.14: Diagrama de Clases de la Clase *Técnico*.

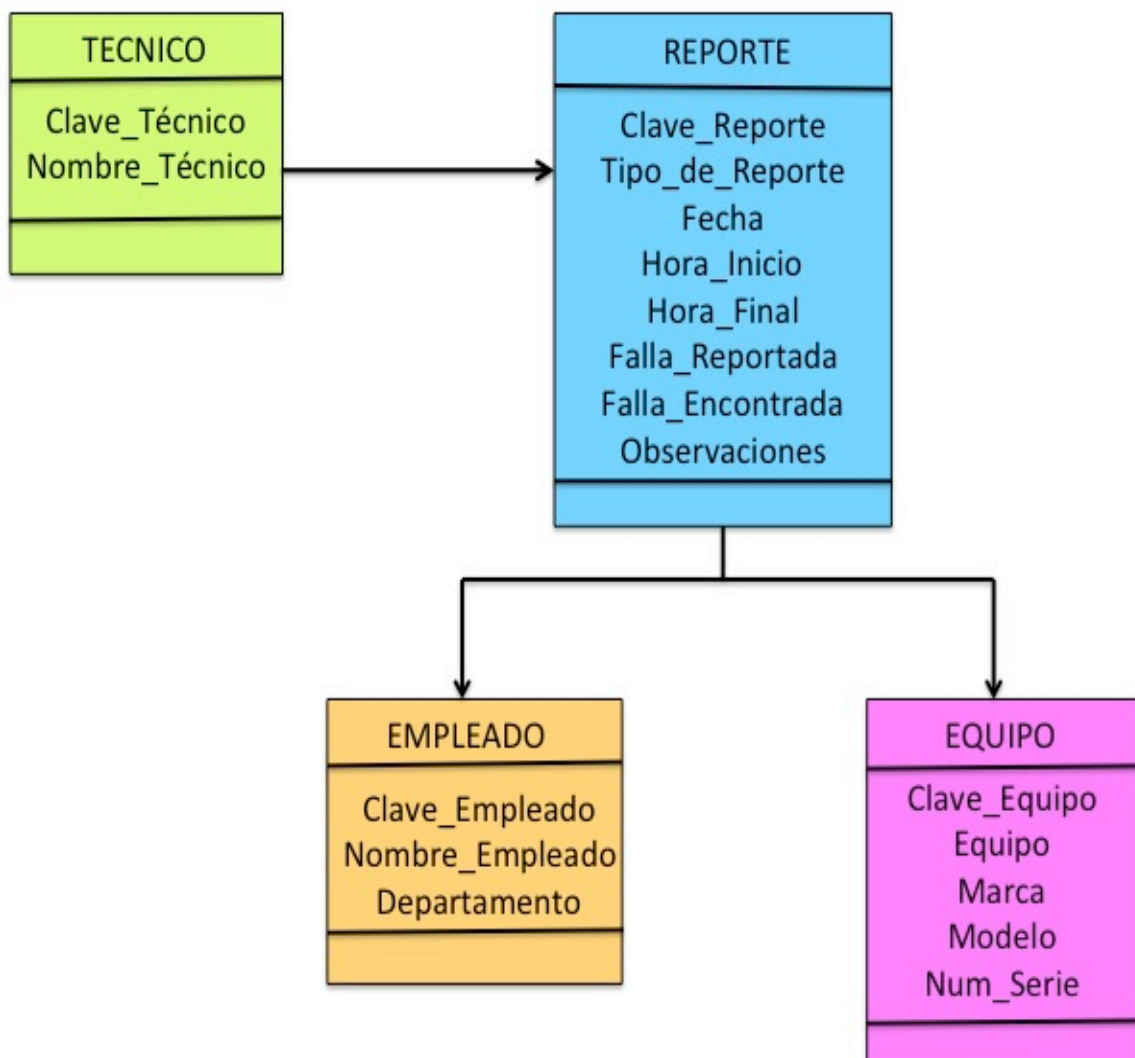
Diagrama de Clases:

Figura 5.15: Diagrama de Clases de la Aplicación *iShelper_App*.

CAPITULO No. 6: “DESARROLLO DE LAS APLICACIONES: iShelperApp Y iShelperForMac”

6.1 10 Sencillos Pasos para Desarrollar una Aplicación para Dispositivos Móviles:

La mejor forma de aprender a desarrollar Aplicaciones para el iPhone, iPod Touch y el iPad es la práctica constante, por ejemplo: realizar un proyecto de desarrollo de cualquier tema que uno elija y que sea de nuestro agrado personal.

En primer lugar, para que este método funcione correctamente, y no nos produzca una completa frustración, tenemos que ser muy realistas con la Aplicación que queremos desarrollar.

El Realismo se entiende que cuando apenas vamos a entrar en este ambiente del Desarrollo de Aplicaciones para Dispositivos Móviles, así como también desarrollar Sistemas de Información para las Computadoras Mac (iMac, MacBook, MacBook Pro y MacBook Air), no podemos empezar con proyectos enormes, en donde lo único que hallaremos es que como no conocemos el Lenguaje de Programación, así como las Sintaxis de las declaraciones, eso conlleva a que tenemos que desechar esa idea, la cual nos lleva a sentirnos mal, ya que nos hace falta tener más conocimientos acerca de la programación y el entorno en general de este tema, que es de reciente creación.

A continuación, se mencionan los 10 Pasos Básicos para realizar cualquier proyecto de Desarrollo de Aplicaciones para los Dispositivos Móviles de Apple, Inc. (iPhone, iPod Touch y el iPad):

1. La Idea, sencilla pero útil:

Cabe mencionar que es importante que el proyecto sea sencillo para que se pueda realizar en poco tiempo, pero también que sea útil, para que nuestro trabajo no finalice realizando acciones que no se encuentren establecidas en la finalidad de nuestro proyecto.

2. Retos Técnicos:

Si en el proyecto que vayamos a desarrollar se va a utilizar el acelerómetro, la aplicación *Mail*, acceder a la biblioteca Música del dispositivo, y no sabemos exactamente cómo se hace, primero tenemos que identificar cuántos y cuáles son esos "retos". No se puede empezar a programar y luego vayan surgiendo las dificultades con este tipo de retos. Además, el tiempo empleado en superar estos retos, será de gran utilidad, no solo para este proyecto, sino también para los proyectos que se desarrollen en el futuro.

3. Soluciones Técnicas:

Para encontrar una buena cantidad de información acerca de la Programación de Dispositivos Móviles, hay que dedicarle el suficiente tiempo para lograr los objetivos propuestos, el cual es estudiar los ejemplos que encontremos, su sintaxis, para llevar a cabo otros ejercicios prácticos que sean similares para ver su funcionalidad con la ayuda del Simulador Virtual del Dispositivo Móvil, para que con esos elementos, podamos seleccionar cuáles son los más adecuados que se puedan utilizar en el proyecto que estemos desarrollando.

4. La Interfaz:

Es de suma importancia que la Interfaz sea amigable con el usuario, útil, y sobre todo, que sea práctica. En eso se caracterizan Apple, Inc., y las aplicaciones del iPhone, iPod touch y el iPad por la sencillez y practicidad de las mismas. En algunas de las aplicaciones cuenta con los botones necesarios para su buen funcionamiento.

También hay que tener en cuenta un aspecto que es importante y básico al mismo tiempo, el cual es el tamaño de la pantalla, el cual se tiene que aprovechar al máximo posible. Para eso, se recomienda seguir las Instrucciones que da Apple, Inc., en sus “**Human Interface Guidelines**”.

5. Apariencia y Utilidad:

En este punto se llega a la conclusión de que es más importante que las aplicaciones funcionen correctamente, de acuerdo a su programación en el código fuente de los archivos “Cabecera” (.h) e “Implementación” (.m), ya que si esto no se realiza correctamente, de nada sirve que el diseño de la interfaz sea “*agradable hacia el público*”, ya que esto conlleva a la frustración de la persona que se encuentra desarrollando la aplicación.

6. Diseño Técnico de la Aplicación:

Un aspecto positivo que tiene Objective-C, es que una forma u otra obliga a prever lo que se va a hacer antes de hacerlo. Por ejemplo, si se quiere un método que permita a un determinado botón realizar una acción, primero se tiene que incluir en el archivo “Cabecera” (.h), y luego programarlo en el archivo “Implementación” (.m). Esto significa que hay que analizar cuáles van a ser las operaciones que va a realizar la aplicación, además de los componentes de la interfaz que son los *IBOutlet*s y cuáles no, así como también cuántos controladores de vista (*View Controllers*) se van a requerir, etc., y dedicar el tiempo necesario para crear las clases y programar las Cabeceras de las mismas.

7. Llenar los Vacíos:

En este punto existen una serie de métodos definidos que no tienen código dentro, y lo que se tiene que hacer ahora es rellenar los huecos con el código correspondiente para que realice su función específica.

8. Pruebas:

La recomendación más importante es que hay que estar probando constantemente las aplicaciones, ya que cada actualización de la misma para la corrección de bugs puede tardar en publicarse en la “*App Store*” entre 3 y 20 días. Se recomienda probar todas las funcionalidades de la aplicación con “*mala idea*”. Una vez que se haya superado este Plan de Pruebas, hay que instalar la aplicación en el dispositivo móvil, para que se les pueda enseñar a familiares y/o amigos para que la prueben como funciona, y así puedan dar su comentario acerca de la aplicación.

9. Publicar la Aplicación:

Si consideramos que la aplicación es una más o menos buena, que ha sido diseñada e implementada por nuestra parte, sometida a pruebas y que inclusive, la cataloguemos como un éxito, habrá que pensar cuidadosamente la lista de precios en la cual habría que fijarle una determinada cantidad de dinero, o incluso, si uno quiere que la aplicación sea de forma gratuita.

10. Cuidar a los Clientes:

Ya sea que paguen o no, las mejores recomendaciones que pueda tener nuestra aplicación, son los Clientes. Por eso hay que leer con frecuencia las reseñas que hacen acerca de la aplicación, y con eso se pueden corregir los errores e incluso modificar o agregar algunos elementos.

También se recomienda actualizar la aplicación, para que los compradores la actualicen, y la vuelvan a probar, para que así realicen nuevos comentarios acerca de la misma.

6.2 CRUD (ABM):

En Computación, **CRUD** es el acrónimo de: Crear, Obtener, Actualizar y Borrar (*del original en inglés: Create, Read, Update and Delete*). Se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un Sistema de Software.

En algunos lugares, se utilizan las siglas **ABM** para lo mismo (*Alta, Baja y Modificación*), haciendo a un lado la operación de *Obtener*; el acrónimo **ABC** para *Altas, Bajas y Cambios*, **ABML** siendo la letra “L” de *Listar, Listado o Lectura*; o **ABMC** siendo la letra “C” de *Consulta*.

También se utiliza el acrónimo de **ABCDEF**: *Agregar, Buscar, Cambiar, Desplegar (Listar), Eliminar. Fichar (Ficha, Cédula o Reporte de un registro).*

6.3 Desarrollo de la Aplicación “iShelperApp” para el iPad:

La aplicación **iShelperApp** consiste en la generación de reportes de soporte técnico en el iPad, para que posteriormente se envíe la información recabada vía inalámbrica a través del correo electrónico, para que se almacene en una base de datos dentro del Departamento de Soporte Técnico de la Empresa.

Esta aplicación se va a desarrollar de la misma forma que las aplicaciones de tipo **ABM** (*Alta, Baja y Modificación*), ya que tiene las mismas operaciones para almacenar, modificar y/o eliminar los datos recabados con anterioridad.



Figura 6.1: Aplicación de tipo ABM (Alta, Baja y Modificación).

La aplicación **iShelperApp** se va a realizar de acuerdo a los siguientes pasos:

- Crear la aplicación (App).
- Modelo de datos y Controladores necesarios.
- Crear una Clase *Store* siguiendo el patrón “*Singleton*”.
- Construcción del “*Storyboard*” de la aplicación.
- Listado de reportes.

- Alta de nuevos reportes.
- Detalle de un reporte.
- Modificación de un reporte.
- Listado de reportes en modo edición.
- Persistencia de los datos a disco.
- Envío de la Información utilizando el correo electrónico.

A continuación, se explica con detalle cada uno de los pasos antes mencionados:

Crear la aplicación (App):

Para el desarrollo de la aplicación se va a utilizar la plantilla “*Single View Application*” de Xcode.

Enseguida, se crea un nuevo proyecto en Xcode con los siguientes valores:

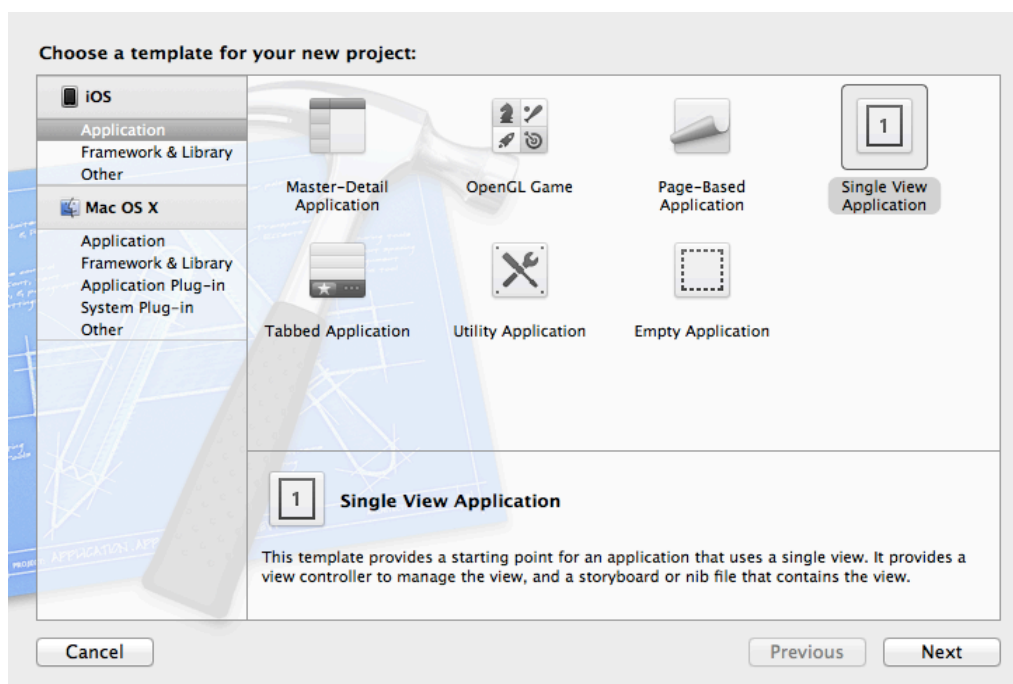


Figura 6.2: Creación de un nuevo Proyecto en Xcode 4.

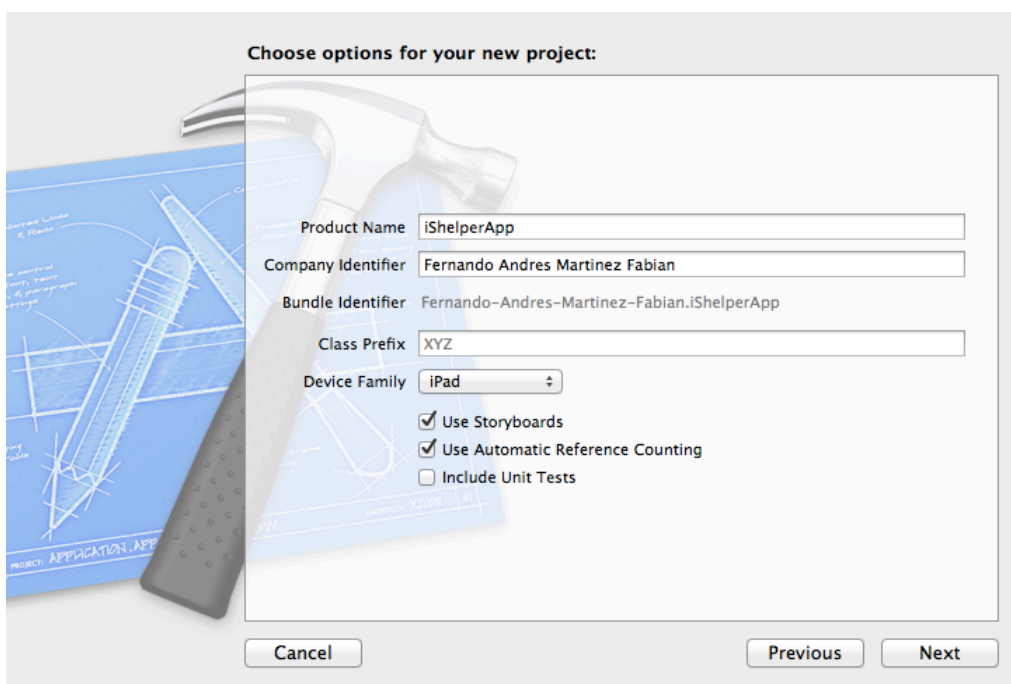


Figura 6.3: Asignación del Nombre del nuevo Proyecto en Xcode 4.

Modelo de Datos y Controladores necesarios:

Como se trata de un proyecto tipo “ABM” (*Alta, Baja y Modificación*), se van a utilizar nomenclaturas lo más generales posibles.

Primero se van a crear todas las clases que se van a necesitar durante la realización de este proyecto:

- iShelperAppltem (subclase de NSObject).
- iShelperAppltemStore (subclase de NSObject).
- iShelperAppAltaViewController (subclase de UIViewController).
- iShelperAppDetalleViewController (subclase de UIViewController).
- iShelperAppEditarViewController (subclase de UIViewController).

Lo único que se va a modificar por el momento, va a ser el contenido de la Clase “*iShelperAppltem*”. Esta Clase sirve para representar lo que va a ser la entidad de negocio de la aplicación. En un ejemplo de la vida cotidiana, la Clase se llamaría Tarea, Receta, Contacto, etc.

En esta Clase “*iShelperAppltem*”, se van a incluir catorce propiedades, las cuales se utilizarán más adelante:

iShelperAppltem.h

```

@property (nonatomic, copy) NSString * reportenum;
@property (nonatomic, copy) NSString * tipomantenimiento;
@property (nonatomic, copy) NSString * equipo;
@property (nonatomic, copy) NSString * marca;
@property (nonatomic, copy) NSString * modelo;
@property (nonatomic, copy) NSString * numserie;
@property (nonatomic, copy) NSString * fallareportada;
@property (nonatomic, copy) NSString * fallaencontrada;
@property (nonatomic, copy) NSString * observaciones;
@property (nonatomic, copy) NSString * nombre;
@property (nonatomic, copy) NSString * departamento;
@property (nonatomic, copy) NSString * fecha;
@property (nonatomic, copy) NSString * hora;
@property (nonatomic, copy) NSString * tecnico;

```

Figura 6.4: Lista de las Catorce Propiedades que contiene la Clase *iShelperAppltem.h*.

Ahora, para permitir a Xcode crear los “*getters*” y “*setters*” asociados a las mismas, se deben hacer “*synthesize*” de todos.

iShelperAppltem.m

```

@synthesize reportenum, tipomantenimiento, equipo, marca, modelo, numserie, fallareportada,
fallaencontrada, observaciones, nombre, departamento, fecha, hora, tecnico;

```

Figura 6.5: Realización del *synthesize* de las Catorce Propiedades.

Además de estos elementos, también se tiene dentro del proyecto un “*iShelperAppViewController*”, el cual provee la plantilla de Xcode, y que será el encargado de listar los reportes de la aplicación.

Creación de una Clase “Store”, siguiendo el patrón “Singleton”:

En este proyecto, se va a crear una Clase de tipo “Store” (*almacén*), que implementará el patrón “Singleton”. Este patrón permite restringir la instanciación de objetos de una determinada clase, de tal forma que únicamente exista en la aplicación una única instancia de dicha clase, mediante el uso de una variable local de tipo “*static*”.

Existen otras formas de implementar un “ABM” de este tipo, se puede utilizar el patrón “*Delegate*” o incluso mediante “*Notifications*”.

iShelperAppItemStore.h

```

@interface iShelperAppItemStore : NSObject {
    NSMutableArray *items;
}

+(iShelperAppItemStore *)sharedStore;
-(NSArray *)allItems;

```

Figura 6.6: Creación de una Clase de tipo *Store*, utilizando el Patrón *Singleton* (I).

iShelperAppItemStore.m

```

@implementation iShelperAppItemStore

+ (iShelperAppItemStore *)sharedStore {
    static iShelperAppItemStore *sharedStore = nil;
    if (!sharedStore) {
        sharedStore = [[super allocWithZone:nil] init];
    }
    return sharedStore;
}

+(id)allocWithZone:(NSZone *)zone {
    return [self sharedStore];
}

-(id)init {
    self = [super init];
    if (self) {
        NSString *path = [self itemArchivePath];
        items = [NSKeyedUnarchiver unarchiveObjectWithFile:path];
        if (!items){
            items = [[NSMutableArray alloc] init];
        }
    }
    return self;
}

-(NSArray *)allItems {
    return items;
}

```

Figura 6.6: Creación de una Clase de tipo *Store*, utilizando el Patrón *Singleton* (II).

Para evitar los errores que muestra Xcode, se debe incluir un *“import”* de la clase *“iShelperAppItem”*. Como se puede observar, hay un método de clase *“(+) sharedStore”* el cual devuelve la instancia existente, y en caso de que no exista, automáticamente la crea. Por otro lado en el método *“init”*, se está invocando al *“init”* del *“super”*. Además, se crea un método de instancia *“allItems”* que permite obtener el *“NSMutableArray”* con todos los reportes que tenemos.

Creación de todas las Pantallas de la aplicación en “Storyboarding” (Guión Gráfico):

Para este punto, se va a utilizar una metodología para la creación de *“Storyboards”*, el cual permita trabajar más rápido de ahora en adelante.

Antes de desarrollar una aplicación, se recomienda realizarla en papel o con herramientas de prototipado, lo que viene siendo el esquema de las pantallas que conforman la aplicación, así como la relación entre las mismas y los elementos que componen cada una de ellas. Es decir, el *“Storyboard”* en papel.

Es por eso que aunque sea un proceso complicado de llevar a cabo al principio, con esto se van a crear todos los elementos del *“Storyboard”* desde el principio, ya que permite centrarnos después en el código y *“olvidarse”* de la presentación.

Otra ventaja que se tiene con este método, es que se puede crear únicamente el *“Storyboard”* y guardar el proyecto como ***“Proyecto Base”*** para la realización de futuras aplicaciones. Así, teniendo ya creada la base de la interfaz, sólo se tiene que incorporar la lógica y el código fuente específico para dicha aplicación.

Antes de comenzar y dado que el *“iShelperAppViewController”* se quiere que sea un listado, se debe modificar para que sea una subclase de *“UITableViewController”* e indicar que se quiere ser *“delegate”* y *“datasource”* de dicha clase.

iShelperAppViewController.h

```
@interface iShelperAppViewController : UITableViewController
    <UITableViewDataSource, UITableViewDelegate>

    -(IBAction)nuevo:(id)sender;
```

Figura 6.8: Código Fuente para que la Clase *iShelperAppViewController.h* sea un Listado.

A continuación, en el archivo *“MainStoryboard.storyboard”*, se añaden primero todos los *“View Controllers”* que se van a necesitar en la aplicación.

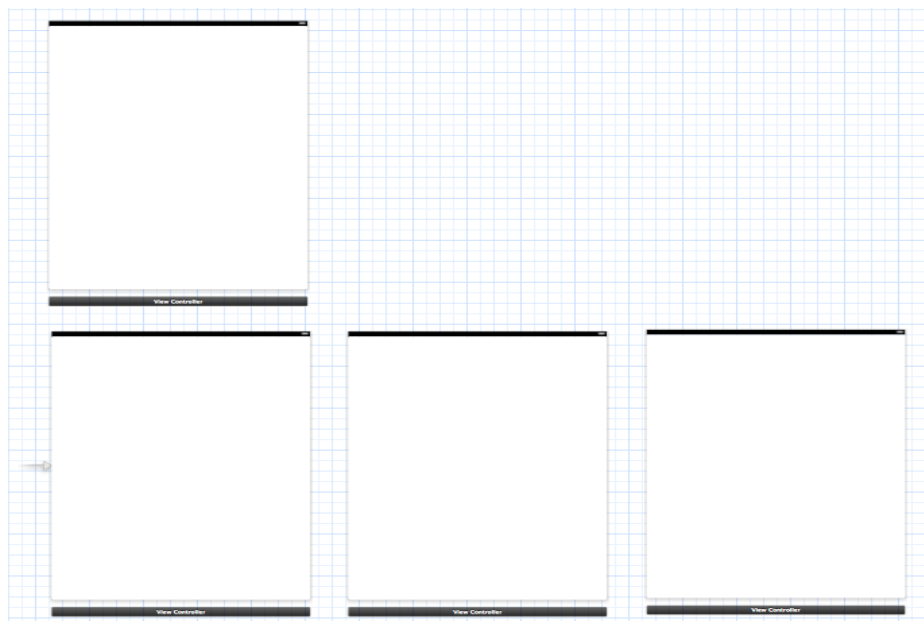


Figura 6.9: Agregación de todos los *View Controllers* que se requieren en la Aplicación.

Una vez que se han añadido todos a la escena, se procede a definir las clases que están asociados a cada uno de ellos (desde el *“Identity Inspector”*).

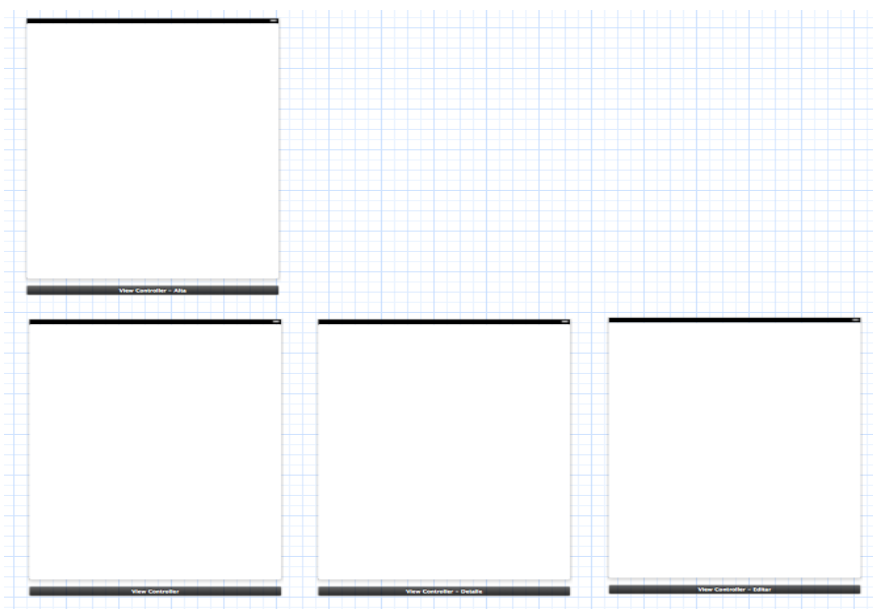


Figura 6.10: Una vez que se han agregado los *View Controllers* a la escena, se definen las Clases que están asociados a cada uno de ellos.

Ahora se van a ir añadiendo los elementos que se requieren en cada uno de ellos.

iShelperAppViewController:

Primero se va a empezar con la escena *“View Controller”*. Se elimina el *“View Controller”* de la escena y se añade un nuevo *“UITableViewController”*, seleccionando en las propiedades como clase asociada *“iShelperAppViewController”*. Como se requiere navegar entre las diferentes pantallas de la aplicación, se necesita un *“Navigation Controller”*. Para eso, teniendo seleccionado el *“View Controller”*, hay que dirigirse al Menú *“Editor” / “Embed In” / “Navigation Controller”*.

A continuación, se añade el botón *“Añadir”*, de tipo *“Bar Button Item”*, en la parte izquierda de la barra de navegación, el cual se utilizará después para dar de alta los nuevos elementos. Se seleccionan en sus propiedades, la opción *“Identifier”* de tipo *“Add”*, el cual hará que Xcode muestre de forma automática el símbolo *“+”*.

Arrastrando desde este nuevo botón a la escena *“Alta View Controller”*, se define el evento del mismo, seleccionando un *“Segue”* de tipo *“Modal”*. Además, en las propiedades del *“Segue”*, se tiene que indicar como identificador *“altatem”*.

En el *“Table View”*, se configura el *“Identifier”* de la celda como *“UITableViewCell”*. Y el *“Style”* de tipo *“Subtitle”*.

Por último, se hace doble click en la cabecera del *“Table View”*, y se pone a continuación el nombre de *“iShelperApp”*.

iShelperAppDetalleViewController:

Primero, se va a enlazar el *“Table View”* anterior con esta ventana. Para eso, se arrastra y se suelta de la celda que se ve en el Table View a la escena *“Detalle View Controller”*, y al soltar se elige la opción *“Push”* (*para hacer uso del “Navigation Controller”*). A este *“Segue”* se debe identificar en la pestaña de propiedades como *“detalleItem”*.

Se añaden a la escena catorce objetos: once *“UILabel”* y tres *“UITextView”*. A los UITextView se deben indicar que no sea editable a través de sus propiedades. La escena quedará como en la siguiente imagen:

Reporte No. Label

Tipo de Mantenimiento: Label

Equipo: Label Marca: Label

Modelo: Label No. de Serie: Label

Falla Reportada:

Falla Encontrada por el Técnico:

Observaciones:

DATOS DEL USUARIO

Empleado: Label Fecha: Label

Area: Label Hora: Label

Técnico: Label

Enviar Datos

I Shelper App Detalle View Controller - Detalles del Reporte

Figura 6.11: Pantalla con los *Detalles del Reporte*.

A continuación, se selecciona la vista *“Assistant Editor”*, se arrastran de uno en uno estos elementos al código para que Xcode crea los *“Outlets”* correspondientes (seleccionando *“Connection”* de tipo *“Outlet”*). Los cuales se denominan a continuación como: **txtTitulo**, **txtTipoMantenimiento**, **txtEquipo**, **txtMarca**, **txtModelo**, **txtNumSerie** (todos ellos son de tipo *“UILabel”*), **txtFallaReportada**, **txtFallaEncontrada**, **txtObservaciones** (los tres son de tipo *“UITextView”*), **txtNombre**, **txtDepartamento**, **txtFecha**, **txtHora** y **txtTecnico** (estos cinco últimos son de tipo *“UILabel”*).

Dado que desde esta pantalla, se va a acceder a la modificación del elemento, se debe incluir un botón en la barra de navegación y crear un *“Segue”* de tipo *“Modal”* con el siguiente controlador, denominado *“Editar View Controller”*. A este *“Segue”* se le nombrará con el identificador *“editarItem”*.

iShelperAppEditarViewController:

Porque se trata de una ventana invocada de forma modal, se puede ver que no cuenta con barra de navegación. Sin embargo, para que siga teniendo homogeneidad con el resto de la aplicación, y se pueden incluir los botones de *“Cancelar”* y *“Guardar”*, se tiene que incluir una *“Navigation Bar”*.

En esta pantalla, se deben incluir los elementos que permitan al usuario final modificar los campos del ítem que se quiera modificar. Para eso, se deben incluir catorce elementos: once *“UITextField”* y tres *“UITextView”*.

De la misma forma en como se hizo en el *“iShelperAppDetalleViewController”*, se debe crear mediante el *“Assistant editor”*, los *“Outlets”* necesarios (*nombrándolos txtTitulo, txtTipoMantenimiento, txtEquipo, txtMarca, txtModelo, txtNumSerie, txtFallaReportada, txtFallaEncontrada, txtObservaciones, txtNombre, txtDepartamento, txtFecha, txtHora y txtTecnico*).

También se deben incluir dos botones (*de tipo “Cancel” y “Done”*) en la barra de navegación que permitan cancelar o guardar los cambios introducidos.

La navegación de ambos botones se va a controlar mediante el código, y por ello no se van a crear *“Segues”* en el *“Storyboard”*, sino que se van a crear un *“IBAction”* para cada uno arrastrando y soltando sobre el ícono del *“ViewController”*. Se van a nombrar estos *“IBActions”*: *“Cancel”* y *“Save”*.

iShelperAltaViewController:

Porque se trata de una ventana invocada de forma modal, se puede ver que no cuenta con barra de navegación. Sin embargo, para que siga teniendo homogeneidad con el resto de la aplicación, y se pueden incluir los botones de “Cancelar” y “Guardar”, se tiene que incluir una “Navigation Bar”.

En esta pantalla, se deben incluir los elementos que permitan al usuario final modificar los campos del ítem que se quiera modificar. Para eso, se deben incluir catorce elementos: once “UITextField” y tres “UITextView”.

De la misma forma en como se hizo en el “iShelperAppDetalleViewController”, se debe crear mediante el “Assistant editor”, los “Outlets” necesarios (nombrándolos *txtTitulo*, *txtTipoMantenimiento*, *txtEquipo*, *txtMarca*, *txtModelo*, *txtNumSerie*, *txtFallaReportada*, *txtFallaEncontrada*, *txtObservaciones*, *txtNombre*, *txtDepartamento*, *txtFecha*, *txtHora* y *txtTecnico*).

También se deben incluir dos botones (de tipo “Cancel” y “Done”) en la barra de navegación que permitan cancelar o guardar los cambios introducidos.

La navegación de ambos botones se va a controlar mediante el código, y por ello no se van a crear “Segue” en el “Storyboard”, sino que se van a crear un “IBAction” para cada uno arrastrando y soltando sobre el ícono del “View Controller”. Se van a nombrar estos “IBActions”: “Cancel” y “Save”.

The image shows a software interface for creating a new maintenance report. The window title is "Nuevo Reporte" and it has "Cancel" and "Save" buttons. The form is set against a yellow background and contains the following fields:

- Reporte No.
- Tipo de Mantenimiento:
- Equipo: Marca:
- Modelo: No. de Serie:
- Falla Reportada:
- Falla Encontrada:
- Observaciones:

DATOS DEL USUARIO

- Empleado: Fecha:
- Area: Hora:
- Técnico:

At the bottom of the screen, there is a dark bar with the text "Alta View Controller".

Figura 6.12: Pantalla con la *Captura de Datos* para generar el nuevo Reporte de Mantenimiento (Preventivo/Correctivo).

Implementar un Table View a Modo de Listado:

En el archivo “*iShelperAppViewController*” ya se tiene un objeto de tipo “*UITableView*” el cual va a servir para presentar al usuario los ítems que contiene la clase “*Store*”.

Para eso, primero se debe verificar que en la clase “*iShelperAppViewController*” se indica que se va a hacer cargo de: “*UITableViewDelegate*” y “*UITableViewDataSource*”.

iShelperAppViewController.h

```
@interface iShelperAppViewController : UITableViewController
    <UITableViewDataSource, UITableViewDelegate>

    -(IBAction)nuevo:(id)sender;
```

Figura 6.13: Código Fuente para implementar un Table View en Modo de Listado.

A continuación, se van a implementar los métodos requeridos por dichos protocolos.

iShelkperAppViewController.m

```
-(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
    return [[[iShelperAppItemStore sharedStore] allItems] count];
}

-(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"UITableViewCell"];
    if (!cell) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:
            @"UITableViewCell"];
    }

    iShelperAppItem *item = [[[iShelperAppItemStore sharedStore] allItems] objectAtIndex:indexPath.
        row];
    [[cell.textLabel] setText:[item reportenum]];
    [[cell.detailTextLabel] setText:[item tipomantenimiento]];

    return cell;
}
```

Figura 6.14: Implementación de los Métodos requeridos por los Protocolos *UITableViewDelegate* y *UITableViewDataSource*.

Para evitar los errores que aparecen en el código, se deben importar las clases “*iShelperAppItem*” y “*iShelperAppItemStore*”.

Se puede observar en el Método “*cellForRowAtIndexPath*”, se está obteniendo la instancia única de “*iShelperAppItemStore*” e invocando a su Método “*allItems*”, el cual va a devolver la totalidad de elementos que contiene el “*NSMutableArray*” que posee.

Implementar el Alta de Nuevos Elementos en el Table View:

Para dar de alta nuevos elementos se necesitan hacer varias cosas, primero se tiene que implementar dentro de el “*Store*” un Método que permita crear un nuevo elemento en el “*NSMutableArray*” y que lo devuelva para poder rellenarlo a través de los campos en pantalla.

iShelperAppItemStore.h

```
-(iShelperAppItem *)insertItem;
```

Figura 6.15: Implementación del Alta de los nuevos elementos en el Table View de la Clase *iShelperAppItemStore.h*.

iShelperAppItemStore.m

```
-(iShelperAppItem *)insertItem {
    iShelperAppItem *item = [[iShelperAppItem alloc] init];
    [items addObject:item];
    return item;
}
```

Figura 6.16: Implementación del Alta de los nuevos elementos en el Table View de la Clase *iShelperAppItemStore.m*.

Se debe importar la clase “*iShelperAppItem*” para evitar los errores.

A continuación, en el archivo “*iShelperAppAltaViewController*”, se va a implementar la lógica de los Métodos “*Cancel*” y “*Save*”.

iShelperAppViewController.m:

```

-(IBAction)cancel:(id)sender {
    [self dismissModalViewControllerAnimated:YES];
}

-(IBAction)save:(id)sender {
    iShelperAppItem *item = [[iShelperAppItemStore sharedStore] insertItem];
    [item setReportenum:[txtTitulo text]];
    [item setTipomantenimiento:[txtTipoMantenimiento text]];
    [item setEquipo:[txtEquipo text]];
    [item setMarca:[txtMarca text]];
    [item setModelo:[txtModelo text]];
    [item setNumserie:[txtNumSerie text]];
    [item setFallareportada:[txtFallaReportada text]];
    [item setFallaencontrada:[txtFallaEncontrada text]];
    [item setObservaciones:[txtObservaciones text]];
    [item setNombre:[txtNombre text]];
    [item setDepartamento:[txtDepartamento text]];
    [item setFecha:[txtFecha text]];
    [item setHora:[txtHora text]];
    [item setTecnico:[txtTecnico text]];
    [self dismissModalViewControllerAnimated:YES];
}

```

Figura 6.17: Implementación de la Lógica de los Métodos *Cancel* y *Save* dentro de la Clase *iShelperAppAltaViewController.m*.

Para evitar errores se deben importar las clases “*iShelperAppItem*” y “*iShelperAppItemStore*”.

En ambos casos, se finaliza con el cierre de la ventana modal.

Llegando a este punto se ejecuta la aplicación, uno se va a dar cuenta de que al regresar de “*iShelperAppAltaViewController*”, habiendo verificado los datos en la pantalla, el listado del archivo “*iShelperAppViewController*” no presenta los cambios. En este punto se pueden aplicar diferentes criterios, y para este proyecto, se va a hacer de forma muy sencilla.

Se va a implementar el método “*viewWillAppear*” del archivo “*iShelperAppViewController*” para que refresque los datos del “*UITableView*” siempre antes de mostrarlo.

iShelperAppViewController.m

```

-(void)viewWillAppear:(BOOL)animated {
    [[self tableView] reloadData];
}

```

Figura 6.18: Implementación del Método *viewWillAppear* dentro de la Clase *iShelperAppViewController.m*.

Llegado a este punto, se pulsa “*Compile and Run*” y se ve que en el Listado ahora permite añadir los nuevos ítems sin problema alguno, y el listado los muestra al pulsar el botón “*Guardar*”.

Implementar la Consulta de un Elemento:

Para que se puedan recuperar los datos del registro seleccionado en pantalla dentro del archivo “*iShelperAppDetalleViewController*”, primero se le debe enviar a dicho controlador el ítem seleccionado por el usuario.

Dado que no se tiene nada que se puede almacenar esta información, se debe crear en el archivo “*iShelperAppDetalleViewController*” una propiedad para que pueda almacenarlo.

iShelperAppDetalleViewController.h

```
@property (weak, nonatomic) IBOutlet iShelperAppItem *itemSeleccionado;
```

Figura 6.19: Implementación de la Propiedad para que pueda almacenar la información en la Clase *iShelperAppDetalleViewController.h*.

iShelperAppDetalleViewController.m

```
@synthesize itemSeleccionado;
```

Figura 6.20: Implementación de la Propiedad para que pueda almacenar la información en la Clase *iShelperAppDetalleViewController.m*.

A continuación, se va a implementar el método “*prepareForSegue*”:

iShelperAppViewController.m

```
-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if ([segue.identifier isEqualToString:@"detalleItem"]) {
        iShelperAppDetalleViewController *detalleViewController = [segue destinationViewController];
        NSInteger index = [[[self tableView] indexPathForSelectedRow] row];
        iShelperAppItem *item = [[[iShelperAppItemStore sharedStore] allItems] objectAtIndex:index];
        [detalleViewController setItemSeleccionado:item];
    }
}
```

Figura 6.21: Implementación del Método *prepareForSegue*.

Para evitar errores, se debe importar la clase “*iShelperAppDetalleViewController*”.

Es importante que se compruebe con este método que el “Segue” que se está ejecutando es el correspondiente al identificador “detalleItem”.

El siguiente paso es que en el archivo “iShelperAppDetalleViewController” se carguen los datos de sus propiedades en los elementos de la interfaz. Para eso, se va a implementar el método “viewWillAppear”.

iShelperAppDetalleViewController.m

```

-(void)viewWillAppear:(BOOL)animated {

    //Cargamos los valores de BO al cargar la pantalla.
    [txtTitulo setText:[itemSeleccionado reportenum]];
    [txtTipoMantenimiento setText:[itemSeleccionado tipomantenimiento]];
    [txtEquipo setText:[itemSeleccionado equipo]];
    [txtMarca setText:[itemSeleccionado marca]];
    [txtModelo setText:[itemSeleccionado modelo]];
    [txtNumSerie setText:[itemSeleccionado numserie]];
    [txtFallaReportada setText:[itemSeleccionado fallareportada]];
    [txtFallaEncontrada setText:[itemSeleccionado fallaencontrada]];
    [txtObservaciones setText:[itemSeleccionado observaciones]];
    [txtNombre setText:[itemSeleccionado nombre]];
    [txtDepartamento setText:[itemSeleccionado departamento]];
    [txtFecha setText:[itemSeleccionado fecha]];
    [txtHora setText:[itemSeleccionado hora]];
    [txtTecnico setText:[itemSeleccionado tecnico]];

}

```

Figura 6.22: Implementación del Método *viewWillAppear* en *iShelperAppDetalleViewController.m*, para que se carguen los Datos de sus Propiedades en los elementos de la Interfaz.

Se ejecuta la aplicación para comprobar que al pulsar sobre una celda del “Table View”, se accede a la pantalla de detalle correspondiente.

Implementar la Modificación de un Elemento:

Al igual que se hizo en el punto anterior, primero hay que ver el archivo “iShelperAppEditarViewController” le llegue la información correspondiente al ítem, cuyo detalle esta viendo el usuario.

Se debe crear en “iShelperAppEditarViewController” una propiedad para que pueda almacenarlo.

iShelperAppEditarViewController.h

```
@property (weak, nonatomic) IBOutlet iShelperAppItem *itemSeleccionado;
```

Figura 6.23: Implementación de la Propiedad que almacene la información correspondiente al ítem cuyo detalle está viendo el usuario en *iShelperAppEditarViewController.h*.

iShelperAppEditarViewController.m

```
@synthesize itemSeleccionado;
```

Figura 6.24: Implementación de la Propiedad que almacene la información correspondiente al ítem cuyo detalle está viendo el usuario en *iShelperAppEditarViewController.m*.

A continuación, se va a implementar el método “*prepareForSegue*”:

iShelperAppDetalleViewController.m

```
-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    iShelperAppEditarViewController *editarViewController = [segue destinationViewController];
    [editarViewController setItemSeleccionado:itemSeleccionado];
}
```

Figura 6.25: Implementación del Método *prepareForSegue*.

Para evitar errores, se debe importar la clase “*iShelperAppEditarViewController.h*”.

A continuación, hay que asegurar de que los campos donde el usuario realizará las modificaciones recojan los valores necesarios al cargar la pantalla:

iShelperAppEditarViewController.m

```
-(void)viewWillAppear:(BOOL)animated {
    // Cargamos los datos del item seleccionado para editar.
    [txtTitulo setText:[itemSeleccionado reportenum]];
    [txtTipoMantenimiento setText:[itemSeleccionado tipomantenimiento]];
    [txtEquipo setText:[itemSeleccionado equipo]];
    [txtMarca setText:[itemSeleccionado marca]];
    [txtModelo setText:[itemSeleccionado modelo]];
    [txtNumSerie setText:[itemSeleccionado numserie]];
    [txtFallaReportada setText:[itemSeleccionado fallareportada]];
    [txtFallaEncontrada setText:[itemSeleccionado fallaencontrada]];
    [txtObservaciones setText:[itemSeleccionado observaciones]];
    [txtNombre setText:[itemSeleccionado nombre]];
    [txtDepartamento setText:[itemSeleccionado departamento]];
    [txtFecha setText:[itemSeleccionado fecha]];
    [txtHora setText:[itemSeleccionado hora]];
    [txtTecnico setText:[itemSeleccionado tecnico]];
}
```

Figura 6.26: Revisión de los campos en donde el usuario realizará las modificaciones que recojan los valores necesarios al cargar la pantalla.

Por último, se debe implementar el código correspondiente a los métodos “Save” y “Cancel” que se crearon previamente, para guardar los cambios incluidos por el usuario en caso necesario, y volver a la pantalla de detalle.

iShelperAppEditarViewController.m

```

-(void)save:(id)sender {
    [itemSeleccionado setReportenum:[txtTitulo text]];
    [itemSeleccionado setTipomantenimiento:[txtTipoMantenimiento text]];
    [itemSeleccionado setEquipo:[txtEquipo text]];
    [itemSeleccionado setMarca:[txtMarca text]];
    [itemSeleccionado setModelo:[txtModelo text]];
    [itemSeleccionado setNumserie:[txtNumSerie text]];
    [itemSeleccionado setFallareportada:[txtFallaReportada text]];
    [itemSeleccionado setFallaencontrada:[txtFallaEncontrada text]];
    [itemSeleccionado setObservaciones:[txtObservaciones text]];
    [itemSeleccionado setNombre:[txtNombre text]];
    [itemSeleccionado setDepartamento:[txtDepartamento text]];
    [itemSeleccionado setFecha:[txtFecha text]];
    [itemSeleccionado setHora:[txtHora text]];
    [itemSeleccionado setTecnico:[txtTecnico text]];
    [self dismissModalViewControllerAnimated:YES];
}

-(void)cancel:(id)sender {
    [self dismissModalViewControllerAnimated:YES];
}

```

Figura 6.27: Implementación de los Métodos Save y Cancel.

Llegando a este punto se ejecuta la aplicación se debe poder pulsar el botón *Editar*, se cambien los valores del ítem, y al pulsar *Guardar*, se puedan ver los cambios reflejados en *iShelperAppDetalleViewController*.

Modo Edición en el Table View:

Lo que se va a hacer ahora es implementar el *Modo Edición* del *Table View* de tal forma que se puedan obtener las siguientes funcionalidades:

- Permitir al usuario borrar los registros del *Table View* emdiante el gesto de *Swipe* hacia la derecha.
- Permitir al usuario modificar el orden y eliminar el registro con el típico botón *Editar* y *Hecho*.

Antes que nada, se debe implementar en la clase *Store* los métodos que permiten modificar el orden de los registros del *NSMutableArray* y eliminar elementos del mismo.

iShelperAppItemStore.h

```
-(void)deleteItem:(iShelperAppItem *)item;
-(void)changeOrderFrom:(NSInteger)from To:(NSInteger)to;
```

Figura 6.28: Implementación de los Métodos que permiten modificar el orden de los registros del *NSMutableArray* y eliminar elementos del mismo en el *Store* de la Clase *iShelperAppItemStore.h*.

iShelperAppItemStore.m

```
-(void)deleteItem:(iShelperAppItem *)item {
    [items removeObjectIdenticalTo:item];
}

-(void)changeOrderFrom:(NSInteger)from To:(NSInteger)to {
    if (to != from) {
        id obj = [items objectAtIndex:from];
        [items removeObjectAtIndex:from];
        if (to >= [items count]) {
            [items addObject:obj];
        } else {
            [items insertObject:obj atIndex:to];
        }
    }
}
```

Figura 6.29: Implementación de los Métodos que permiten modificar el orden de los registros del *NSMutableArray* y eliminar elementos del mismo en el *Store* de la Clase *iShelperAppItemStore.m*.

En segundo lugar, se va a permitir al usuario borrar los registros con el gesto, y para eso se debe implementar el método “*commitEditingStyle*”:

iShelperAppViewController.m

```

-(void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)
editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath {
    iShelperAppItem *item = [[[iShelperAppItemStore sharedStore] allItems] objectAtIndex:indexPath
row]];
    [[iShelperAppItemStore sharedStore] deleteItem:item];
    [tableView reloadData];
}

```

Figura 6.30: Implementación del método *commitEditingStyle*, el cual permite al usuario borrar los registros.

Y por último, se va a activar el “Modo Edición”, modificando el método “*viewDidLoad*” del archivo “*iShelperAppViewController*”:

iShelperAppViewController.m

```

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.rightBarButtonItem = self.editButtonItem;
}

```

Figura 6.31: Activación del modo *Edición*, modificando el Método *viewDidLoad* en la Clase *iShelperAppViewController.m*.

Con esta sencilla sentencia, el “*Table View*” va a gestionar automáticamente los estados e incluirá todos los elementos gráficos y botones que permiten modificar el orden de elementos o eliminarlos.

Ahora se van a implementar dos métodos de “*UITableView*” que se requieren para aplicar los cambios de orden y eliminación de elementos:

iShelperAppViewController.m

```

-(BOOL)tableView:(UITableView *)tableView canMoveRowAtIndexPath:(NSIndexPath *)indexPath
{
    return YES;
}

-(void)tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath *)sourceIndexPath
toIndexPath:(NSIndexPath *)destinationIndexPath {
    [[iShelperAppItemStore sharedStore] changeOrderFrom:[sourceIndexPath row] To:
[destinationIndexPath row]];
}

```

Figura 6.32: Implementación de dos Métodos de *UITableView* que se requieren para aplicar los cambios de orden y eliminación de los elementos.

El primero de los dos métodos sirve para indicarle al Sistema si se quiere que una fila pueda moverse de su sitio o no. En este caso se devuelve “YES” siempre, pero habrá otros casos en que solamente se interese mantener fija una celda en concreto.

El segundo método es llamado cuando el usuario mueve una fila, y por tanto hay que invocar al método de el “Store” para que realice el cambio.

Se ejecuta la aplicación y se comprueba que ya se puede cambiar el orden de las filas, eliminar las filas tanto desde el modo “Edición”, como haciendo el gesto de “Swipe” hacia la derecha, etc.

Guardar los Datos a Disco:

Una de las cosas que hace falta por hacer es que los registros del “NSMutableArray” se conserven entre una ejecución y otra de la aplicación.

Para eso, se va a guardar la información del “NSMutableArray” a disco mediante “Archiving”. Existen otras opciones, como “Core Data”, “Property Lists”, etc., pero para un volumen no superior a 1,000 registros, no se requiere algo más.

Para poder guardar un objeto “NSMutableArray” en disco, tanto dicha clase, así como también todas las correspondientes a sus contenidos, se deben implementar “NSCoding”. “NSMutableArray” ya lo implementa, pero la clase “iShelperAppltem”.

Por eso, se tienen que modificar los siguientes elementos:

iShelperAppltem.h

```
@interface iShelperAppItem : NSObject <NSCoding>
```

Figura 6.33: Modificación de los elementos NSObject y NSCoder.

Además, se deben implementar dos métodos los cuales permiten codificar y decodificar las diferentes propiedades que tiene el archivo “iShelperAppltem”.

iShelperAppltem.m

```

-(void)encodeWithCoder:(NSCoder *)aCoder {
    [aCoder encodeObject:reportenum forKey:@"Reporte No. "];
    [aCoder encodeObject:tipomantenimiento forKey:@"Tipo de Mantenimiento"];
    [aCoder encodeObject:equipo forKey:@"Equipo"];
    [aCoder encodeObject:marca forKey:@"Marca"];
    [aCoder encodeObject:modelo forKey:@"Modelo"];
    [aCoder encodeObject:numserie forKey:@"NumSerie"];
    [aCoder encodeObject:fallareportada forKey:@"Falla Reportada"];
    [aCoder encodeObject:fallaencontrada forKey:@"Falla Encontrada"];
    [aCoder encodeObject:observaciones forKey:@"Observaciones"];
    [aCoder encodeObject:nombre forKey:@"Nombre"];
    [aCoder encodeObject:departamento forKey:@"Departamento"];
    [aCoder encodeObject:fecha forKey:@"Fecha"];
    [aCoder encodeObject:hora forKey:@"Hora"];
    [aCoder encodeObject:tecnico forKey:@"Tecnico"];
}

-(id)initWithCoder:(NSCoder *)aDecoder {
    self = [super init];
    if (self) {
        [self setReportenum:[aDecoder decodeObjectForKey:@"Reporte No. "]];
        [self setTipomantenimiento:[aDecoder decodeObjectForKey:@"Tipo de Mantenimiento"]];
        [self setEquipo:[aDecoder decodeObjectForKey:@"Equipo"]];
        [self setMarca:[aDecoder decodeObjectForKey:@"Marca"]];
        [self setModelo:[aDecoder decodeObjectForKey:@"Modelo"]];
        [self setNumserie:[aDecoder decodeObjectForKey:@"No. de Serie"]];
        [self setFallareportada:[aDecoder decodeObjectForKey:@"Falla Reportada"]];
        [self setFallaencontrada:[aDecoder decodeObjectForKey:@"Falla Encontrada"]];
        [self setObservaciones:[aDecoder decodeObjectForKey:@"Observaciones"]];
        [self setNombre:[aDecoder decodeObjectForKey:@"Nombre"]];
        [self setDepartamento:[aDecoder decodeObjectForKey:@"Departamento"]];
        [self setFecha:[aDecoder decodeObjectForKey:@"Fecha"]];
        [self setHora:[aDecoder decodeObjectForKey:@"Hora"]];
        [self setTecnico:[aDecoder decodeObjectForKey:@"Tecnico"]];
    }
    return self;
}

```

Figura 6.34: Implementación de dos Métodos, los cuales permiten codificar y decodificar las diferentes Propiedades que tiene la Clase *iShelperAppltem.m*.

Ahora que la clase ya permite ser guardada, se va a implementar en el “Store”, un método que permita guardar su “*NSMutableArray*” a disco:

iShelperApplItemStore.h

```
-(BOOL)saveChanges;
```

Figura 6.35: implementación del Método que permita guardar el *NSMutableArray* a disco en la Clase *iShelperApplItemStore.h*.

iShelperApplItemStore.m

```
-(NSString *)itemArchivePath {
    NSArray *documentDirectories = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
        NSUserDomainMask, YES);
    NSString *documentDirectory = [documentDirectories objectAtIndex:0];
    return [documentDirectory stringByAppendingPathComponent:@"items.archive"];
}

-(BOOL)saveChanges {
    NSString *path = [self itemArchivePath];
    return [NSKeyedArchiver archiveRootObject:items toFile:path];
}
```

Figura 6.36: implementación del Método que permita guardar el *NSMutableArray* a disco en la Clase *iShelperApplItemStore.m*.

Además, se debe hacer que cuando se instancie el “Store”, recupere en caso de no haberlo hecho ya, los datos del disco. Para eso, se modifica el método “*init*” para sustituir el código que se tenía, en donde se crearon los elementos por defecto:

iShelperAppItemStore.m

```

-(id)init {
    self = [super init];
    if (self) {
        NSString *path = [self itemArchivePath];
        items = [NSKeyedUnarchiver unarchiveObjectWithFile:path];
        if (!items){
            items = [[NSMutableArray alloc] init];
        }
    }
    return self;
}

```

Figura 6.37: Modificación del Método *init* para que cuando se instancie la Clase *Store*, recupere los datos del disco.

Por último, se necesita que cuando la aplicación entre en “*background*” grabe a disco los elementos que contiene el “*Store*” en ese momento. Para eso, se implementa el método “*applicationDidEnterBackground*” del “*Delegate*” de la aplicación:

iShelperAppAppDelegate.m

```

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    BOOL success = [[iShelperAppItemStore sharedStore] saveChanges];
    if (success) {
        NSLog(@"Todos los elementos salvados.");
    } else {
        NSLog(@"Ha habido un error al grabar los datos al sandbox.");
    }
}

```

Figura 6.38: Implementación del Método *applicationDidEnterBackground*, para que grabe a disco los elementos que contiene la Clase *Store* en ese momento.

Para evitar errores, se debe importar la clase “*iShelperAppItemStore*”.

Envío de los Reportes de Soporte Técnico a través del Correo Electrónico desde el iPad:

Ahora, se va a mostrar lo fácil que se puede permitir al usuario enviar correos electrónicos desde la aplicación, para que se envíen de forma inalámbrica los Reportes de Soporte Técnico generados desde la aplicación, hacia el Área de Soporte Técnico de la Empresa, para que se pueda llevar a cabo la Toma de Decisiones con respecto al Mantenimiento Correctivo, así como también del Mantenimiento Preventivo.

El SDK de iOS ha hecho que sea muy fácil de enviar correos electrónicos, utilizando los APIs de tipo “built-in”. Con unas cuantas líneas de código, se puede iniciar la interfaz gráfica de correo igual que en la aplicación “Mail” de valores, el cual permite redactar un correo electrónico.

Creando un Botón en la pantalla “Detalle” de la aplicación “iShelperApp”:

Dentro de la pantalla “Detalle”, se va a agregar un botón que diga “Enviar Datos”, el cual se va a encargar de mandar los datos recabados previamente vía inalámbrica al Área de Soporte Técnico.

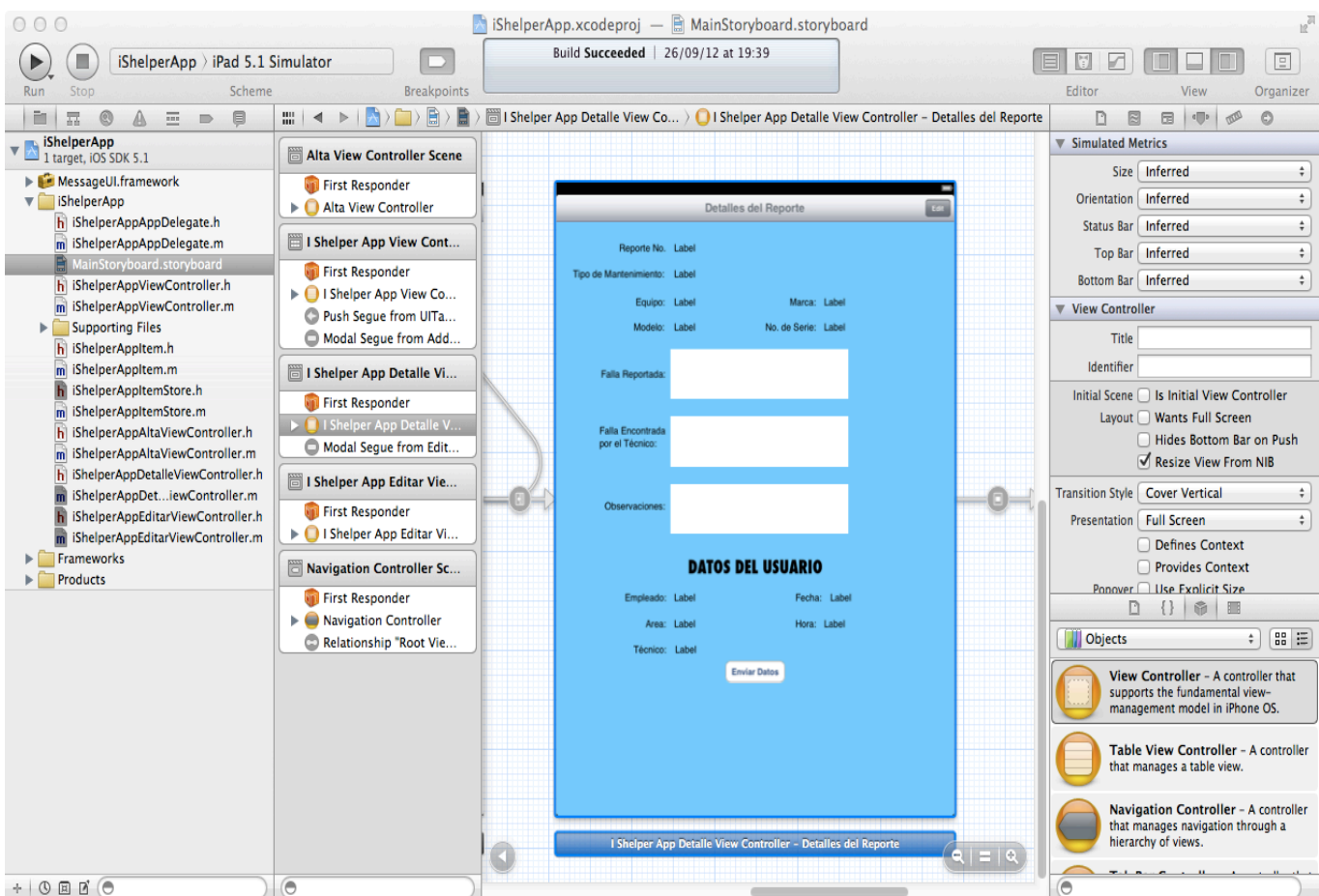


Figura 6.39: Creación de un botón para el envío de la información recabada a través del Correo Electrónico.

La acción que va a realizar la aplicación es que va a aparecer la interfaz del correo electrónico, cuando el usuario pulsa el botón “*Enviar Datos*”.

Implementando la Interfaz del Correo Electrónico:

A continuación, se importa la clase “*MessageUI.h*” y se implementa el Delegate “*MFMailComposeViewControllerDelegate*” en el archivo “*iShelperAppDetalleViewController.h*”.

```
#import <MessageUI/MessageUI.h>
- (IBAction)EnviarDatos:(id)sender;
```

Figura 6.40: Importación de la Clase *MessageUI.h*.

Se selecciona el archivo “*iShelperAppDetalleViewController.m*”, y se implementa el método “*Enviar Datos*”, así como también el “*MFMailComposeViewControllerDelegate*”. Se agrega el siguiente código en el archivo de implementación:

```
- (IBAction)EnviarDatos:(id)sender {
    // Asunto del Correo Electrónico
    NSString *emailTitle = @"Reporte de Soporte Técnico";
    // Contenido del Correo Electrónico
    NSString *messageBody = [NSString stringWithFormat:@"\nEstos son los Datos recabados en el
    Reporte:\n\nTitulo:                %@\n\nMantenimiento:            %@\n\nEquipo:
    %@\n\nMarca:                    %@\n\nModelo:                        %@\n\nNo. de Serie:
    %@\n\nFalla Reportada:          %@\n\nFalla Encontrada:          %@\n\nObservaciones:          %@\n\n
    nNombre:                        %@\n\nDepartamento:          %@\n\nFecha:                %@\n\n
    nHora:                            %@\n\nTécnico:                    %@", txtTitulo.text,
    txtTipoMantenimiento.text, txtEquipo.text, txtMarca.text, txtModelo.text, txtNumSerie.text,
    txtFallaReportada.text, txtFallaEncontrada.text, txtObservaciones.text, txtNombre.text,
    txtDepartamento.text, txtFecha.text, txtHora.text, txtTecnico.text];
    // Para los siguientes Destinatarios
    NSArray *toRecipients = [NSArray arrayWithObject:@"famartinez.fabian@yahoo.com.mx"];

    MFMailComposeViewController *mc = [[MFMailComposeViewController alloc] init];
    mc.mailComposeDelegate = self;
    [mc setSubject:emailTitle];
    [mc setMessageBody:messageBody isHTML:NO];
    [mc setToRecipients:toRecipients];

    // Mostrar el Correo Electrónico en la Pantalla
    [self presentViewController:mc animated:YES completion:NULL];
}

// Dismisses the email composition interface when users tap Cancel or Send. Proceeds to update the
// message field with the result of the operation.
- (void)mailComposeController:(MFMailComposeViewController*)controller didFinishWithResult:
(MFMailComposeResult)result error:(NSError*)error
{
    [self dismissModalViewControllerAnimated:YES];
}
```

Figura 6.41: Implementación del Método *Enviar Datos*, así como también el *MFMailComposeViewControllerDelegate*.

En las líneas 3-20 del código anterior define el asunto del correo electrónico, el contenido del mensaje y los destinatarios. En las líneas (#) se crea y se incorpora el *“MFMailComposeViewController”*. La clase *“MFMailComposeViewController”* proporciona una interfaz estándar que gestiona la edición y envío de un mensaje de correo electrónico. Se puede utilizar este controlador para mostrar una vista de correo electrónico estándar dentro de la aplicación. Se rellenan los campos de esta vista con los valores iniciales, incluyendo el correo electrónico del destinatario, el asunto y el cuerpo del texto.

En la línea 27 del código, se invoca el presente View Controller para mostrar la interfaz de correo en la pantalla.

El *“didFinishWithResult:”* es un método del protocolo *“MFMailComposeViewControllerDelegate”*. Este método será llamado automáticamente cuando la interfaz de correo está cerrada (*por ejemplo, cuando el usuario cancela la operación*). El parámetro resultado le indica el código de resultado, cuando la interfaz de composición electrónica es despedida. En aras de la simplicidad, simplemente se registra el estado en este proyecto. En una aplicación del mundo real, se debe proporcionar un control especial para el código de resultado (*por ejemplo, mostrar un aviso de alerta cuando la aplicación no pueda enviar el correo electrónico*).

Por último, la línea 41 cierra la interfaz de composición electrónica.

Agregando el Framework "MessageUI":

Sin embargo, esto no se realiza todavía. Si se intenta compilar la aplicación, va a mostrar el error de compilación, como se muestra a continuación en la siguiente imagen:

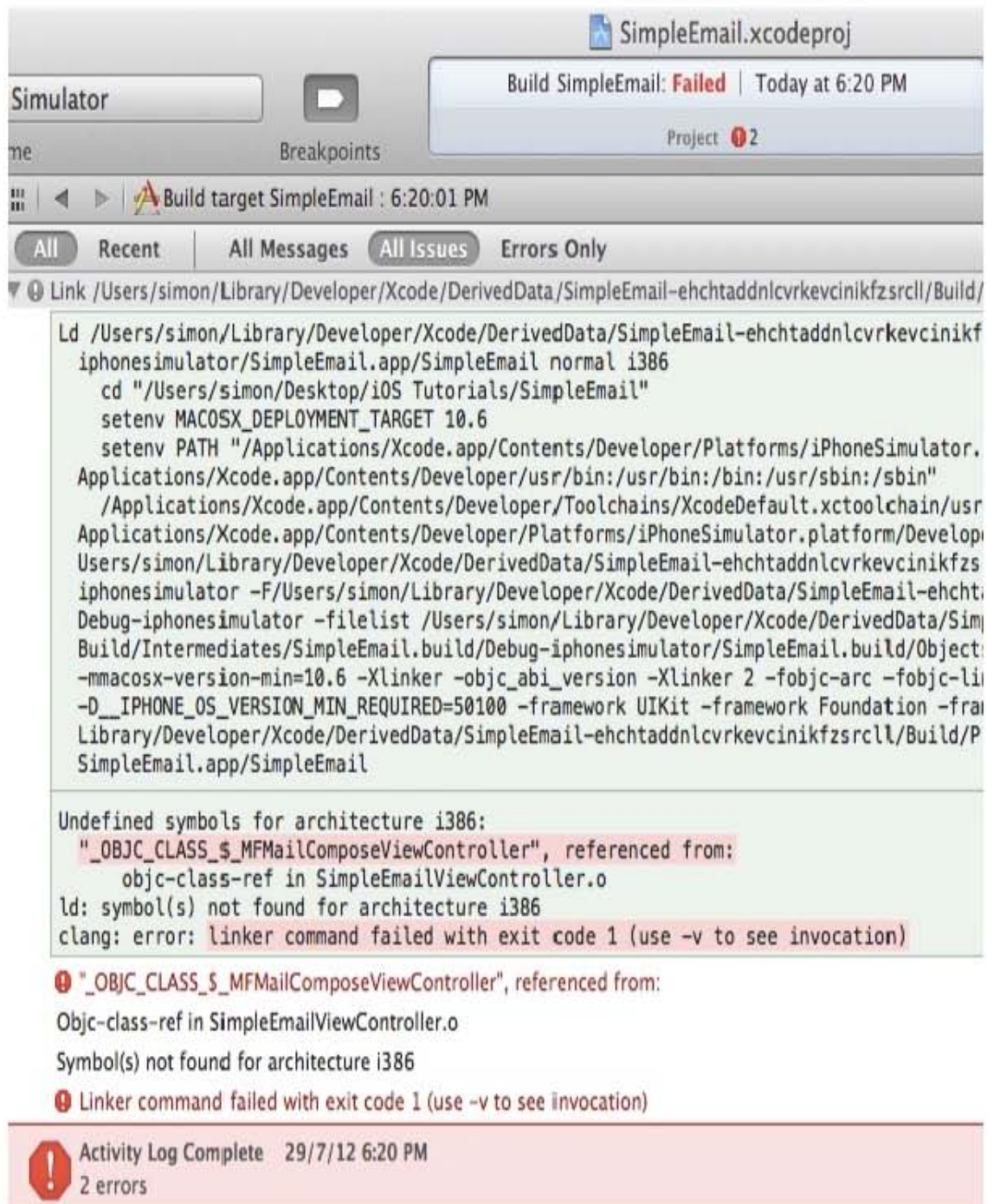


Figura 6.42: Mensaje de Error de Compilación, cuando aún no se ha agregado el Framework *MessageUI*.

El problema es que Xcode no tiene ni idea de lo que es el `MFMailComposeViewController`. Aunque el `MFMailComposeViewController` es un controlador integrado del iOS SDK, es responsabilidad de los desarrolladores, incorporar el framework necesario en el proyecto. Cuando el proyecto de Xcode se crea por primera vez, viene con tres núcleos que incluyen los frameworks UIKit, Foundation y CoreGraphics. La clase `MFMailComposeViewController` está incluida en el framework `MessageUI`, la cual se tiene que agregar manualmente.

Para corregir el error, hay que insertar el framework en el proyecto. En el `Project Navigator`, se selecciona la opción `iShelperApp` en `Targets`. Luego, se da click en `Build Phases` en la parte superior del editor de proyectos. A continuación, se abre la sección `Link Binary With Libraries`.

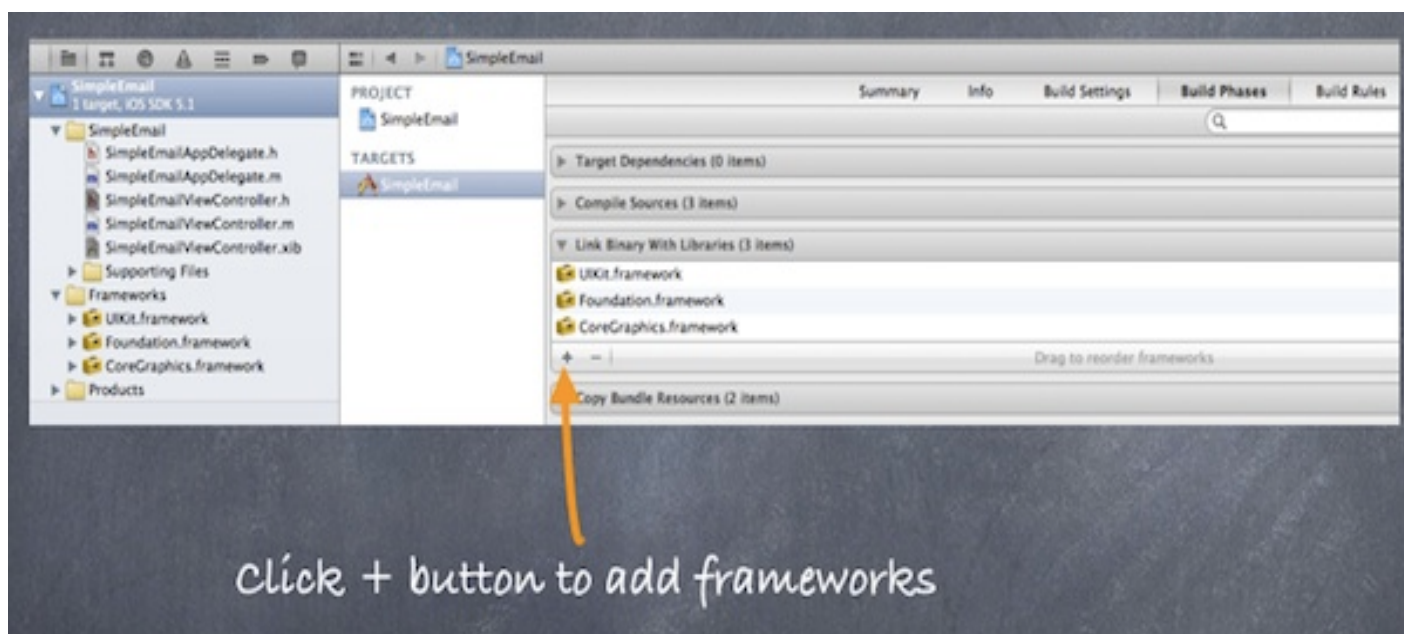
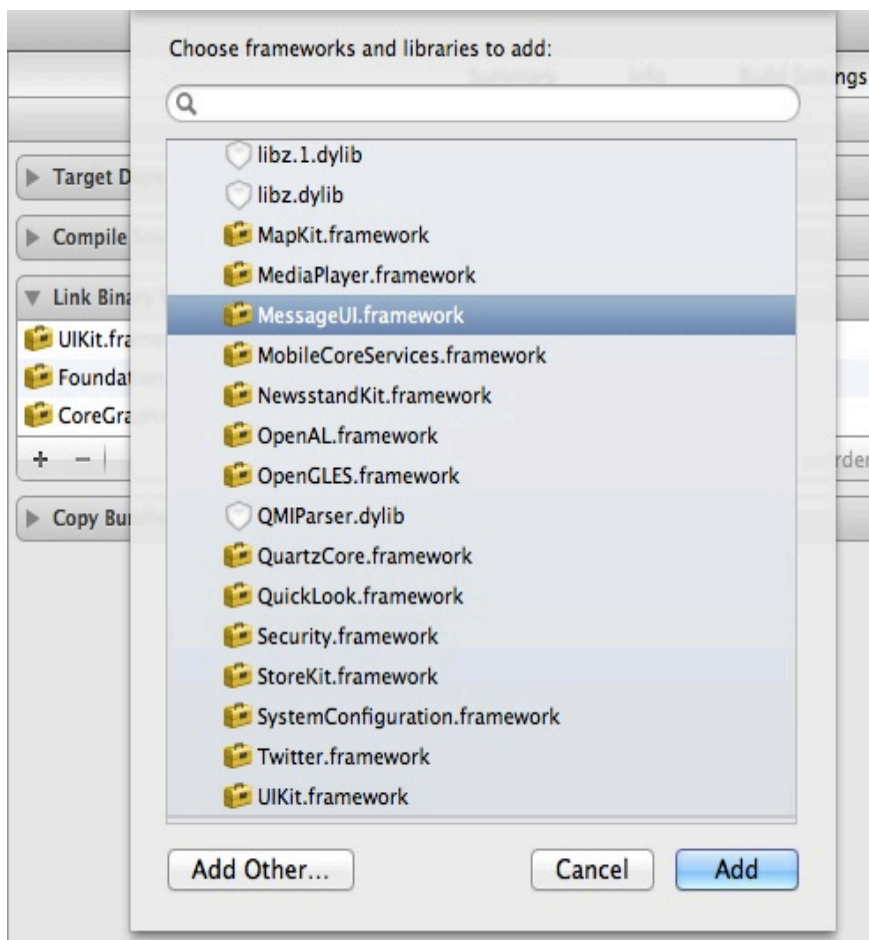


Figura 6.43: Procedimiento para insertar el Framework en el Proyecto.

A continuación, se da click en el botón `+` y se selecciona la opción `MessageUI.framework`. Después, se da click en el botón `Add`, para que Xcode incluya el framework `MessageUI` del proyecto.



6.44: Selección del Framework *MessageUI*.

Ahora bien, el error ya se debe haber corregido. Se ejecuta la aplicación, para ver su correcto funcionamiento. Al tocar el botón “*Enviar Datos*”, se mostrará la pantalla con la interfaz para “*Redactar el Correo Electrónico*” con el contenido básico para hacerlo.

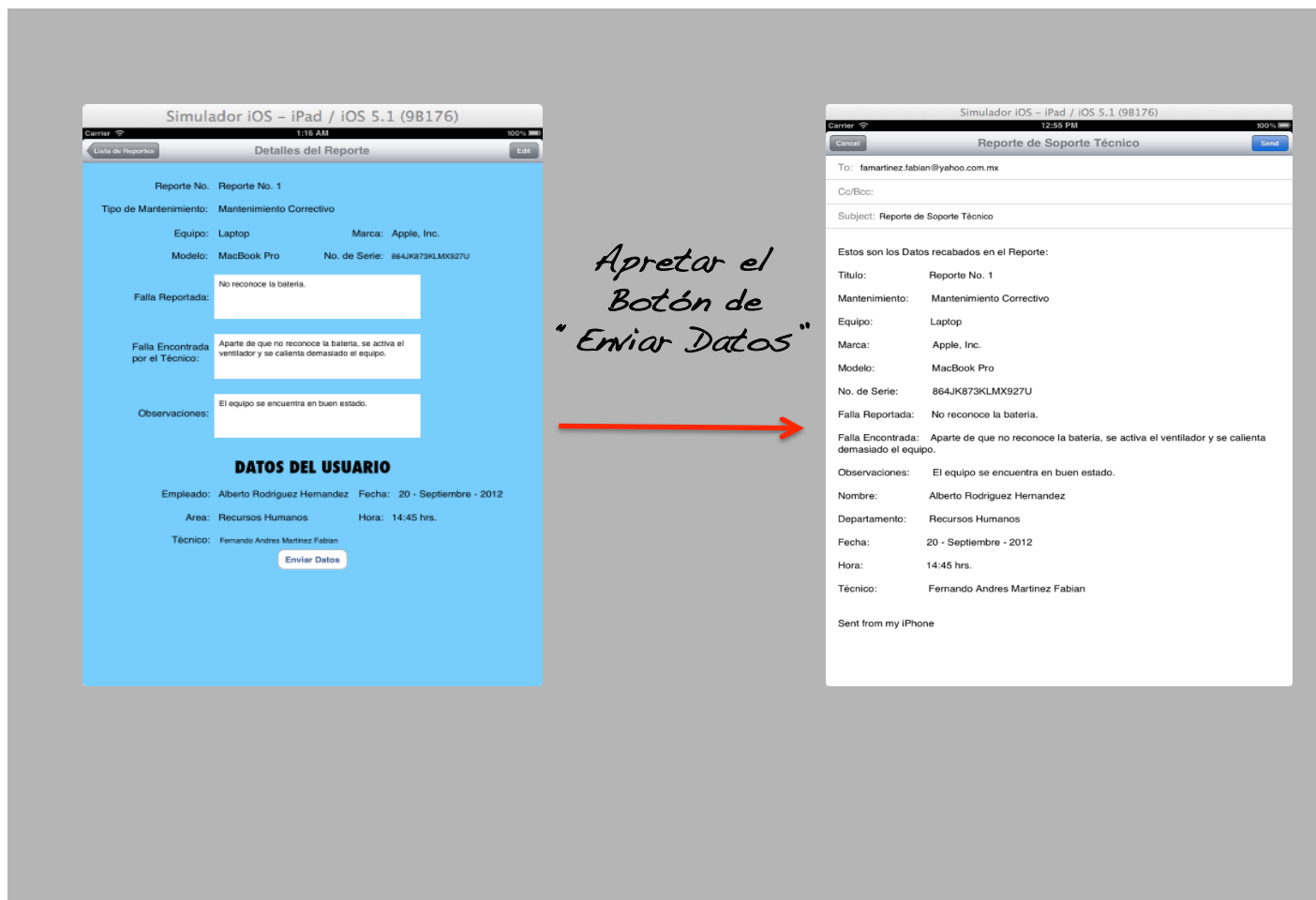


Figura 6.45: Interfaz gráfica del Correo Electrónico con los Datos recabados listos para su envío.

6.4 Desarrollo de la Aplicación “iShelperForMac” para iMac y Macbook:

A continuación, se va a desarrollar la aplicación “iShelperForMac” para utilizarlo en una computadora MacBook o iMac, el objetivo principal es para darle continuidad al tratamiento de la información en el iPad, acerca de los Reportes de Soporte Técnico, esto se hace con la finalidad de llevar un estricto control de los Reportes en el Área de Soporte Técnico de cualquier Empresa, ya sea que esto se realiza de forma mensual, bimestral, semestral y/o anual.

Las Funciones que va a realizar la aplicación “iShelperForMac” son las siguientes:

- Agregar los datos recabados del formulario para mostrarlo en una tabla.
- Eliminar un renglón de la tabla.
- Imprimir los datos recabados de la tabla.

Agregando los Datos en una Tabla y Guardarlos en un Archivo “Plist”:

Lo primero que se tiene que hacer es crear un nuevo proyecto, al cual se le va a asignar el nombre de “*iShelperForMac*”.



Figura 6.46: Pantalla de Bienvenida de Xcode 4.

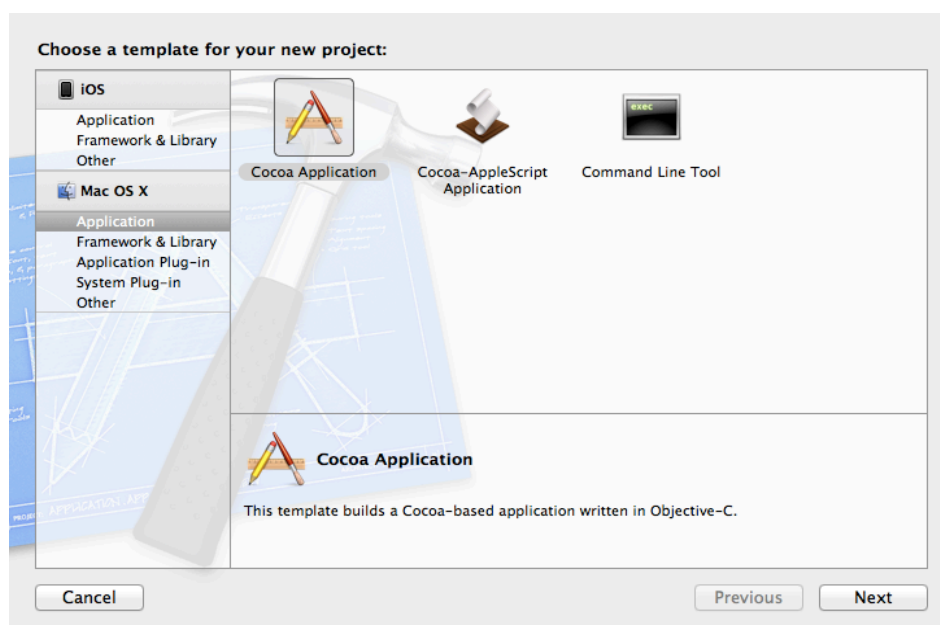


Figura 6.47: Selección de la opción de *Cocoa Application* para MacBook.

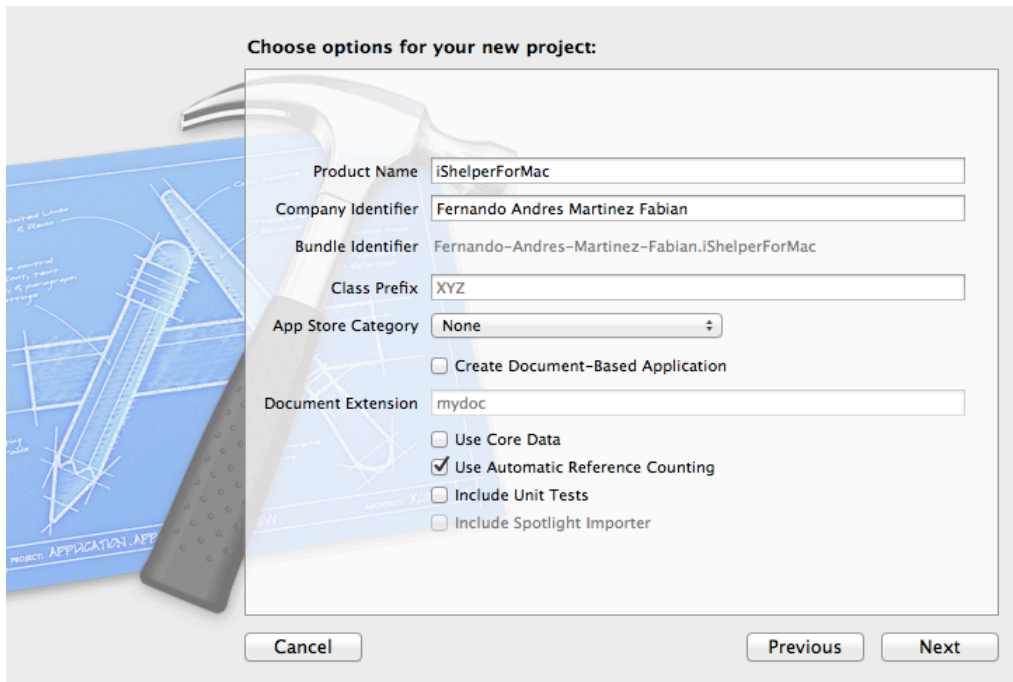


Figura 6.48: Asignación del Nombre del Proyecto.

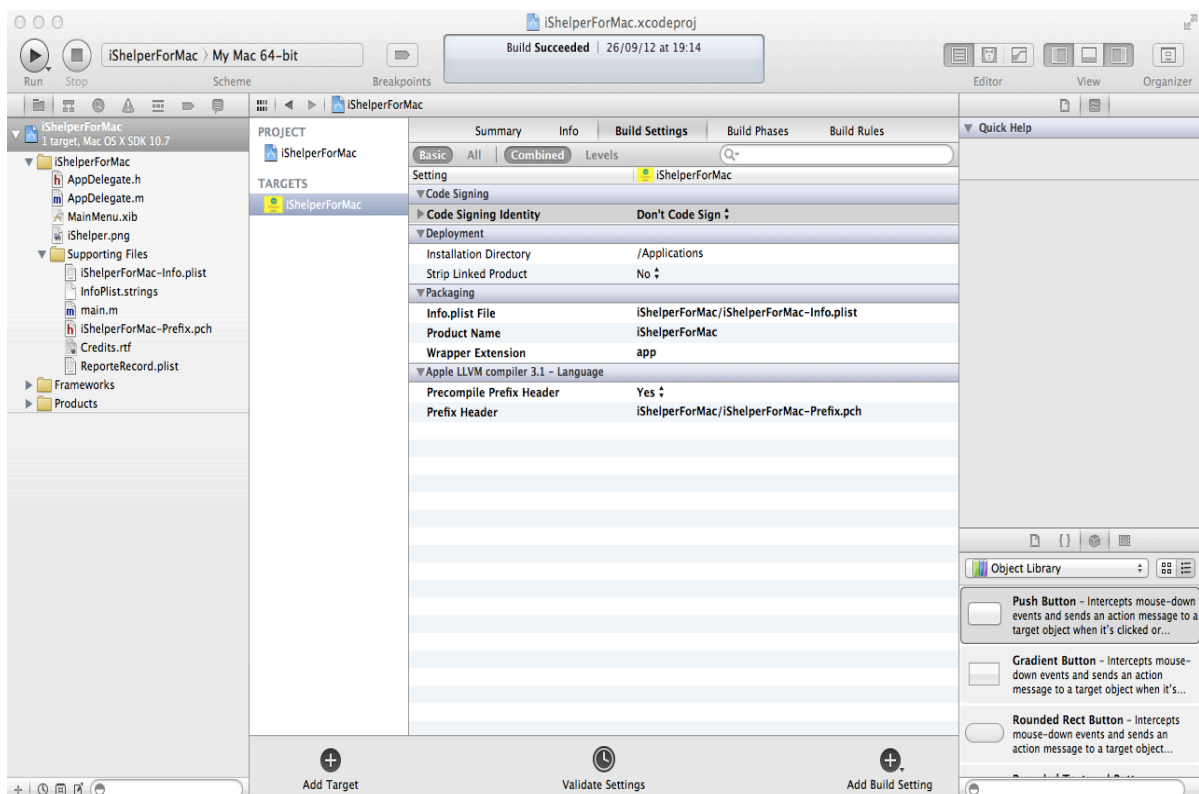


Figura 6.49: Pantalla Principal del nuevo Proyecto creado.

Ahora, en la interfaz gráfica se van a crear los botones, las cajas de texto en donde se van a introducir los datos, la tabla que van a ir en cada pantalla:

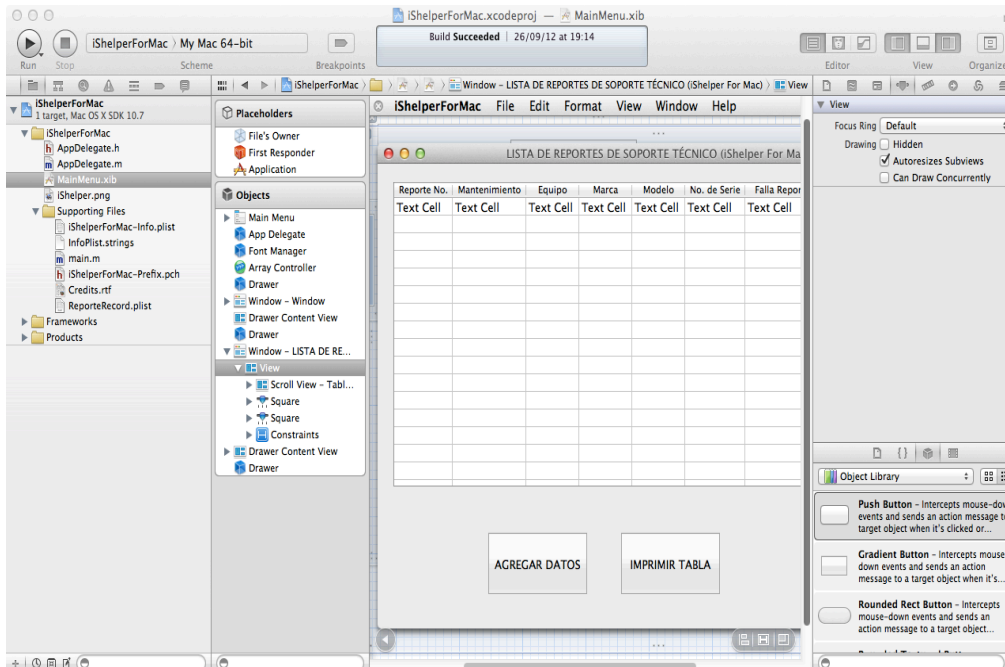


Figura 6.50: Pantalla Principal de la Aplicación, en la cual aparece la tabla en donde se van a ir agregando los datos.

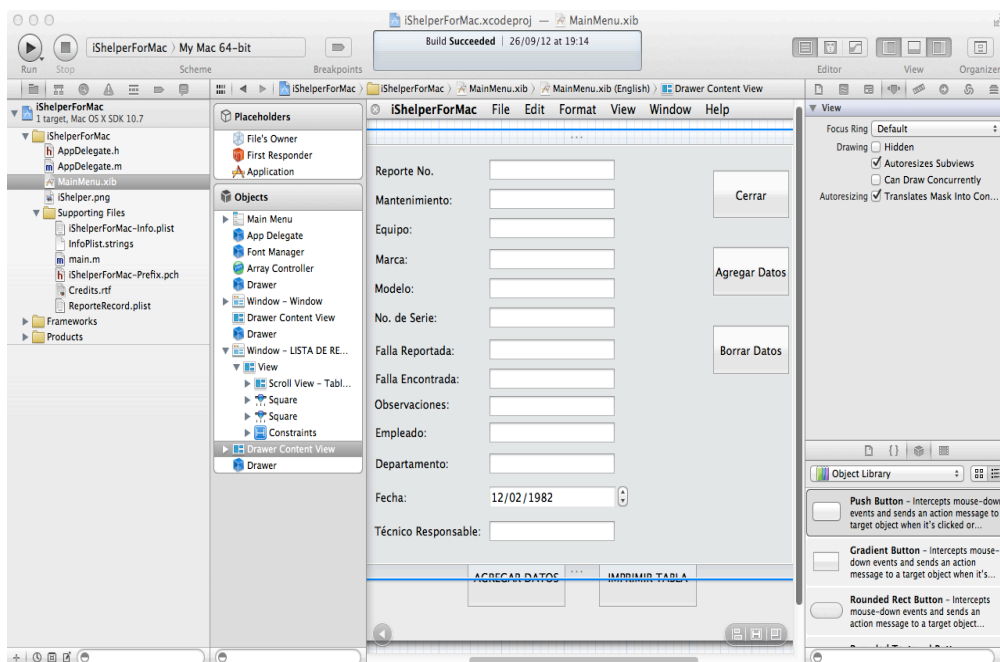


Figura 6.51: Formulario en el cual se van a introducir los datos que se van a mostrar en la Tabla de la Pantalla Principal.

Luego, se escribe a continuación el siguiente código en el archivo “AppDelegate.h”:

```
@interface AppDelegate : NSObject <UIApplicationDelegate>{

    IBOutlet NSArrayController *arrayReportes;

    IBOutlet NSTableView *tblReportes;
    NSMutableArray *reportes;

    IBOutlet NSTextField *txtTitulo;
    IBOutlet NSTextField *txtTipoMantenimiento;
    IBOutlet NSTextField *txtEquipo;
    IBOutlet NSTextField *txtMarca;
    IBOutlet NSTextField *txtModelo;
    IBOutlet NSTextField *txtNumSerie;
    IBOutlet NSTextField *txtFallaReportada;
    IBOutlet NSTextField *txtFallaEncontrada;
    IBOutlet NSTextField *txtObservaciones;
    IBOutlet NSTextField *txtNombre;
    IBOutlet NSTextField *txtDepartamento;
    IBOutlet NSDatePicker *reporteDatePicker;
    IBOutlet NSTextField *txtTecnico;

}

@property (assign) IBOutlet NSWindow *window;
@property (nonatomic, retain) IBOutlet NSDatePicker *reporteDatePicker;
-(IBAction)guardarDatos:(id)sender;
```

Figura 6.52: Introducción del código para guardar los datos recabados previamente, al momento de presionar el botón de *Agregar Datos*.

Después, hay que dirigirse al archivo “AppDelegate.m” para introducir el siguiente código:

```

@synthesize reporteDatePicker;

NSString *reporteFile;

-(IBAction)guardarDatos:(id)sender{

    NSDateFormatter *newFormatter = [[NSDateFormatter alloc] init];

    [newFormatter setDateStyle:NSDateFormatterShortStyle];

    NSString *reporteFechaHora = [newFormatter stringFromDate:reporteDatePicker.dateValue];

    NSDictionary *dictionary = [NSDictionary dictionaryWithObjectsAndKeys:
        [txtTitulo stringValue], @"Reporte No.",
        [txtTipoMantenimiento stringValue], @"Mantenimiento",
        [txtEquipo stringValue], @"Equipo",
        [txtMarca stringValue], @"Marca",
        [txtModelo stringValue], @"Modelo",
        [txtNumSerie stringValue], @"No. de Serie",
        [txtFallaReportada stringValue], @"Falla Reportada",
        [txtFallaEncontrada stringValue], @"Falla Encontrada",
        [txtObservaciones stringValue], @"Observaciones",
        [txtNombre stringValue], @"Empleado",
        [txtDepartamento stringValue], @"Departamento",
        reporteFechaHora, @"Fecha y Hora",
        [txtTecnico stringValue], @"Técnico Responsable",
        nil];

    [arrayReportes addObject:dictionary];

    [txtTitulo setStringValue:@""];
    [txtTipoMantenimiento setStringValue:@""];
    [txtEquipo setStringValue:@""];
    [txtMarca setStringValue:@""];
    [txtModelo setStringValue:@""];
    [txtNumSerie setStringValue:@""];
    [txtFallaReportada setStringValue:@""];
    [txtFallaEncontrada setStringValue:@""];
    [txtObservaciones setStringValue:@""];
    [txtNombre setStringValue:@""];
    [txtDepartamento setStringValue:@""];
    [txtTecnico setStringValue:@""];
}

```

Figura 6.53: Introducción del Código Fuente para que se muestre en la tabla de la Pantalla Principal.

A continuación, se agrega un “*Array Controller*” al proyecto, para que se pueda realizar la conexión del “*arrayReportes*” que se estableció en el código fuente con el “*Array Controller*”.

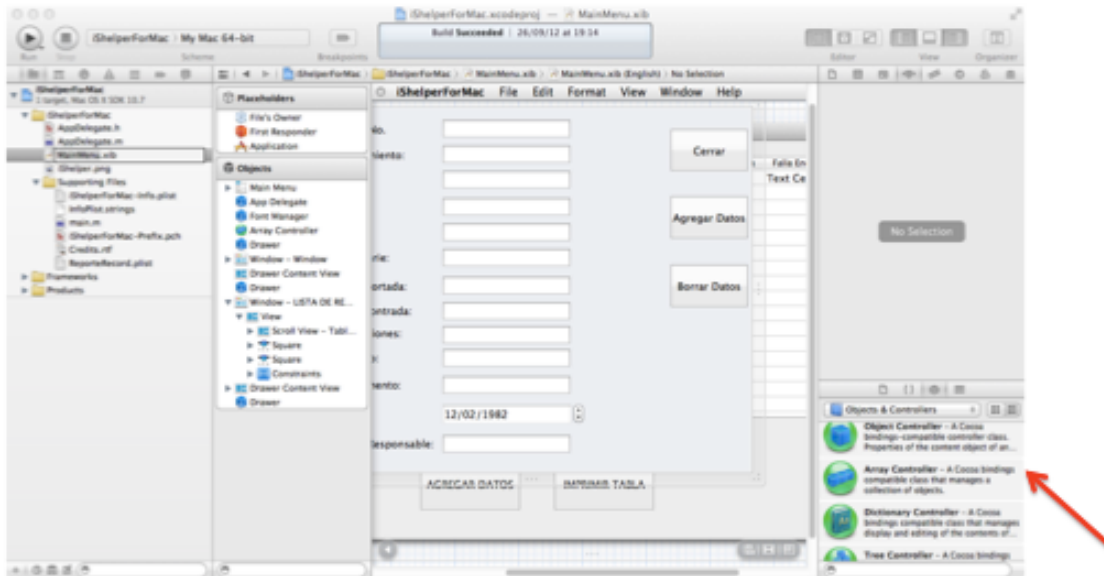


Figura 6.54: Agregación de un Array Controller al Proyecto.

Después en la opción “*Bindings Inspector*”, se selecciona la opción “*Value*”, después se da click en el recuadro junto a donde dice “*Blind to*”, y se escribe el nombre de cada columna en la opción “*Modelo Key Path*”, todo esto se hace con la finalidad de asignar las llaves respectivas.

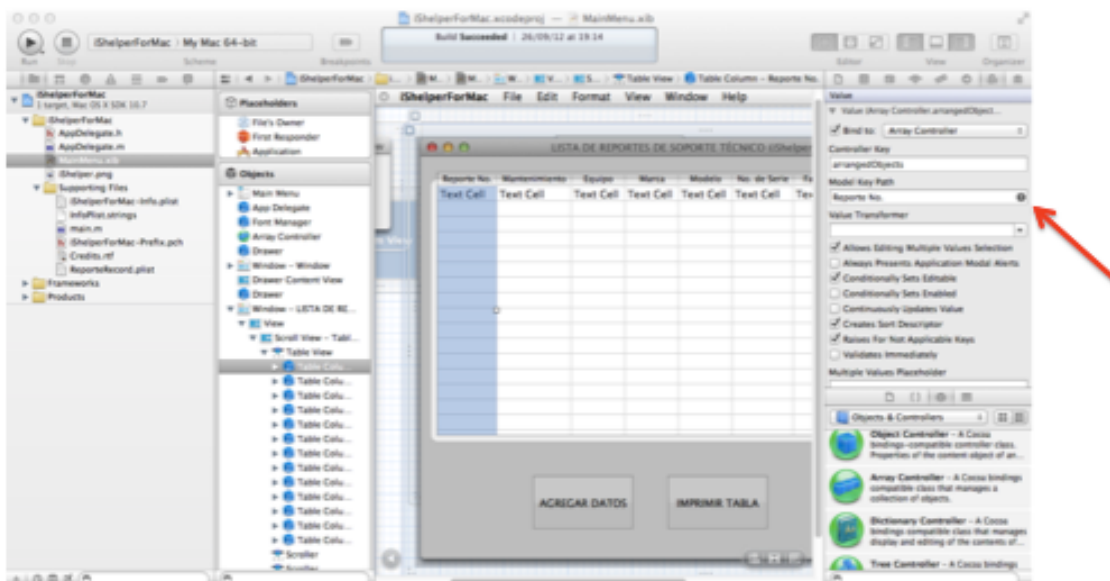


Figura 6.55: Asignación de las Llaves respectivas, mediante la opción *Bindings Inspector*.

Ahora, se realiza la conexión de todos los elementos que contiene el App Delegate con los “*Outlets*” que están en el “*Connection Inspector*”:

Luego, se agrega un botón al cual se le va a poner el nombre de “Agregar Datos”, para que se puedan agregar los datos del formulario en la tabla, así como también se realiza la conexión en el “*Connection Inspector*”.

Después, se ejecuta la aplicación, se insertan los datos, pero el problema es que no se guardan los datos en el disco, porque todavía no se ha agregado el archivo en el cual se pueden guardar los datos, ya que se borran los datos cada vez que se ejecuta la aplicación.

Ahora, para que se puedan guardar los datos que se han ingresado en el formulario, se necesita un archivo con extensión “.plist”, para eso se agrega un “*Property List*”, el cual se encuentra en “*Supporting Files*”, ya que con este archivo al momento de ejecutarse, se agregan los datos a la tabla, se termina de ejecutar, y cuando se vuelve a ejecutar, se puede ver que están presentes los datos que se habían ingresado anteriormente.

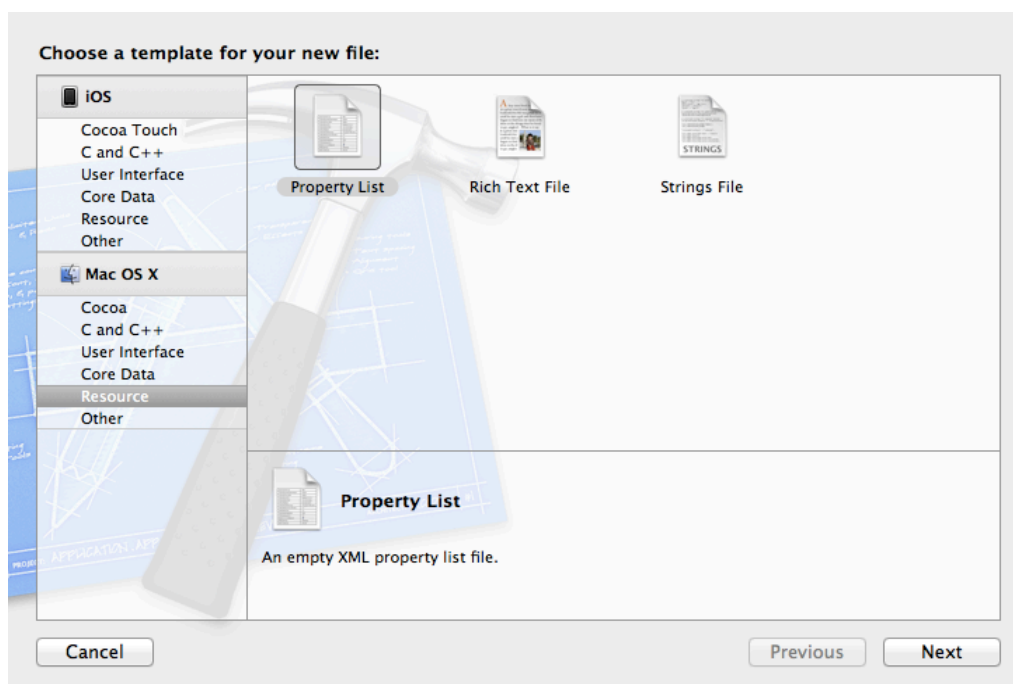


Figura 6.56: Agregación de un *Property List*.

A continuación, se escribe el siguiente código en el archivo “*AppDelegate.m*”:

```
NSString *reporteFile;
```

Figura 6.57: Introducción del Código Fuente para guardar los Datos, al momento de que ingresan en la Tabla (I).

También se escribe el siguiente código, para que junto con el anterior código fuente se pueden guardar los datos que se ingresan en la tabla:

```
if(![[NSFileManager defaultManager] fileExistsAtPath:reporteFile] || [[NSFileManager
defaultManager] isWritableFileAtPath:reporteFile]){

[[arrayReportes arrangedObjects] writeToFile:reporteFile atomically:YES];
}
```

Figura 6.58: Introducción del Código Fuente para guardar los Datos, al momento de que ingresan en la Tabla (II).

Borrar los Datos de un Renglón de la Tabla:

A continuación, lo que se va a hacer es agregar un botón el cual sirve para borrar los datos que se encuentran en la tabla, para esto se realiza dentro del mismo proyecto en el que se está trabajando, en la ventana del formulario, al cual se le va a asignar el nombre de “*Borrar Datos*”, esto se hace mediante el “*Connection Inspector*”, como se muestra en la siguiente imagen:

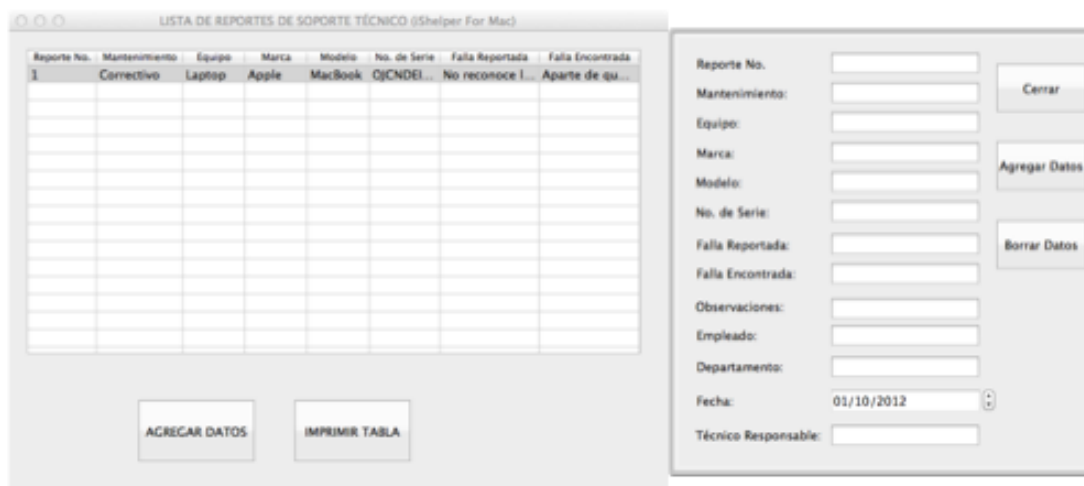


Figura 6.59: Eliminación de un Registro de la Tabla.

Después, se agrega el siguiente código en el archivo “*AppDelegate.h*”, para que realice la función correspondiente de borrar los datos del renglón seleccionado:

```
-(IBAction)borrarDatos:(id)sender;
```

Figura 6.60: Introducción del Código Fuente para eliminar un registro de la Tabla (I).

Y enseguida, se agrega en el archivo “*AppDelegate.m*” el siguiente código fuente:

```

-(IBAction)borrarDatos:(id)sender{
    int row = [tblReportes selectedRow];
    if(row > -1){
        [arrayReportes removeObjectAtIndex:row];
        [tblReportes reloadData];
        [tblReportes deselectAll:nil];
    }
}

```

Figura 6.61: Introducción del Código Fuente para eliminar un registro de la Tabla (II).

Para finalizar, en el “*App Delegate*”, se realiza la conexión de “*borrarDatos*” en la opción de “*Received Actions*” con el botón que se creó en la pantalla del formulario.

Imprimir los Datos Contenidos en una Tabla:

Dentro del mismo proyecto, se va a implementar un botón el cual realice la función de Imprimir los datos contenidos en una tabla, para lo cual es necesario escribir el siguiente código fuente en el archivo “*AppDelegate.m*”:

```

-(IBAction)imprimirArchivo:(id)sender;

```

Figura 6.62: Introducción del Código Fuente para imprimir la Tabla con los registros guardados (I).

Luego, se escribe en el archivo “*AppDelegate.h*” el siguiente código fuente:

```
-(IBAction)imprimirArchivo:(id)sender{
    [tblReportes print:sender];
}
```

Figura 6.63: Introducción del Código Fuente para imprimir la Tabla con los registros guardados (I).

Ahora, se crea el botón en la pantalla principal, al cual se le va a asignar el nombre de “*Imprimir*”, mediante el “*Connection Inspector*”, como se muestra a continuación en la siguiente imagen:

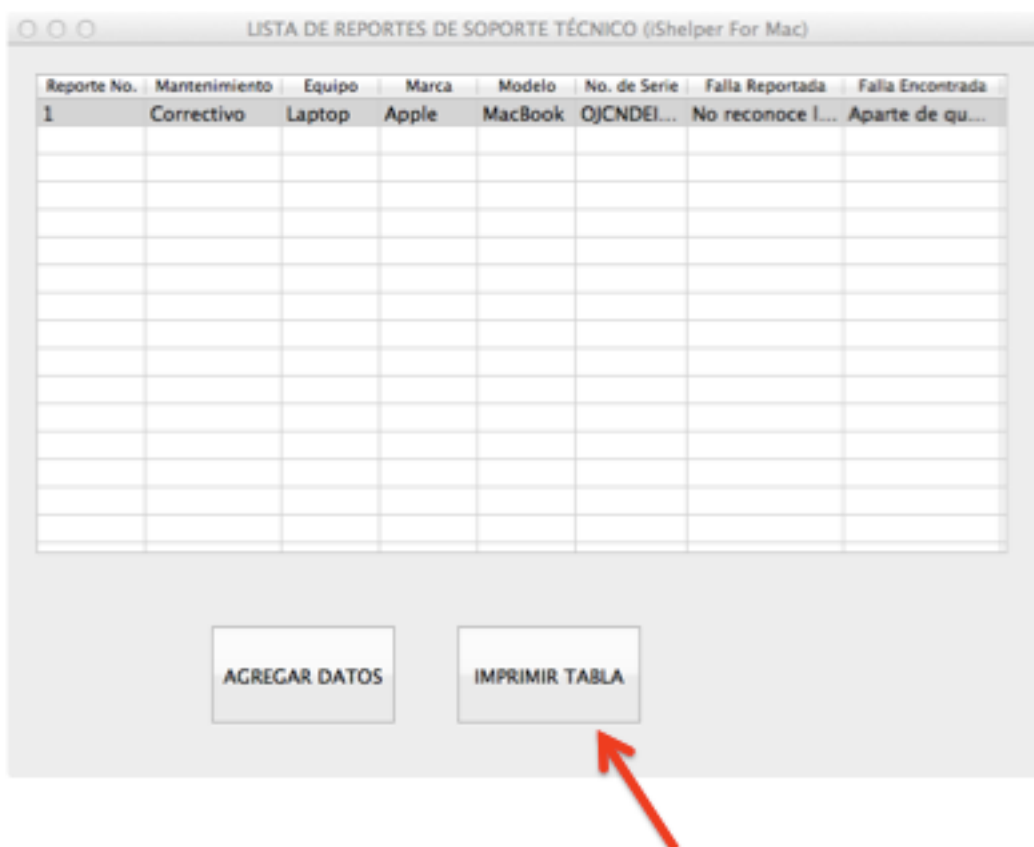


Figura 6.64: Botón *Imprimir Tabla* en la Pantalla principal de la Aplicación.

A continuación, en el “*App Delegate*” se conecta el outlet “*imprimir Archivo*” con el botón “*Imprimir*” que se encuentra en la pantalla principal, y después de que da click en el botón “*Imprimir*” aparece la Vista sin Detalles, y dándole click en el botón “*Show Details*”, el cual muestra la Vista con Detalles para Imprimir la Tabla, como se muestra a continuación en las siguientes imágenes:

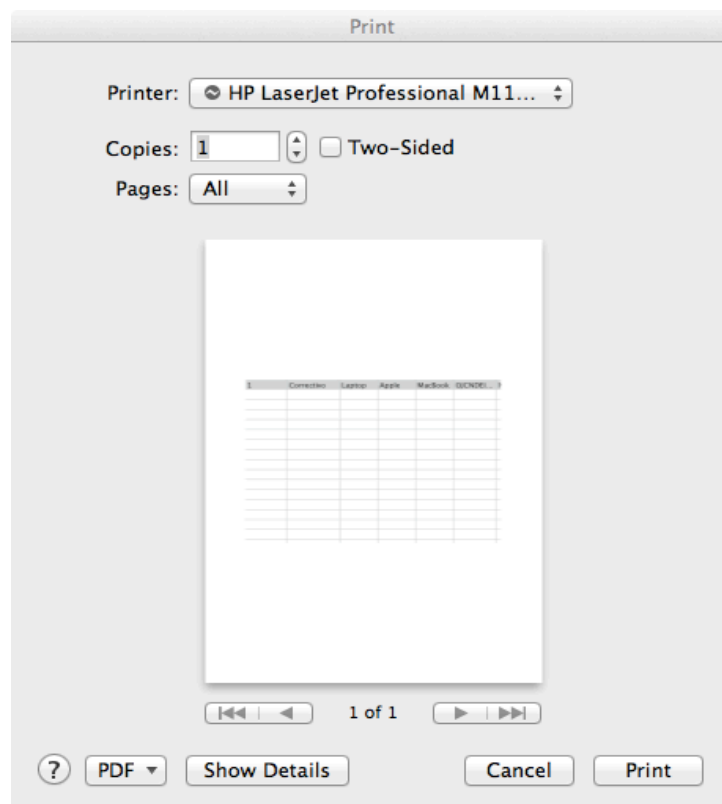


Figura 6.65: Pantalla previa de Impresión (Vista sin Detalles).

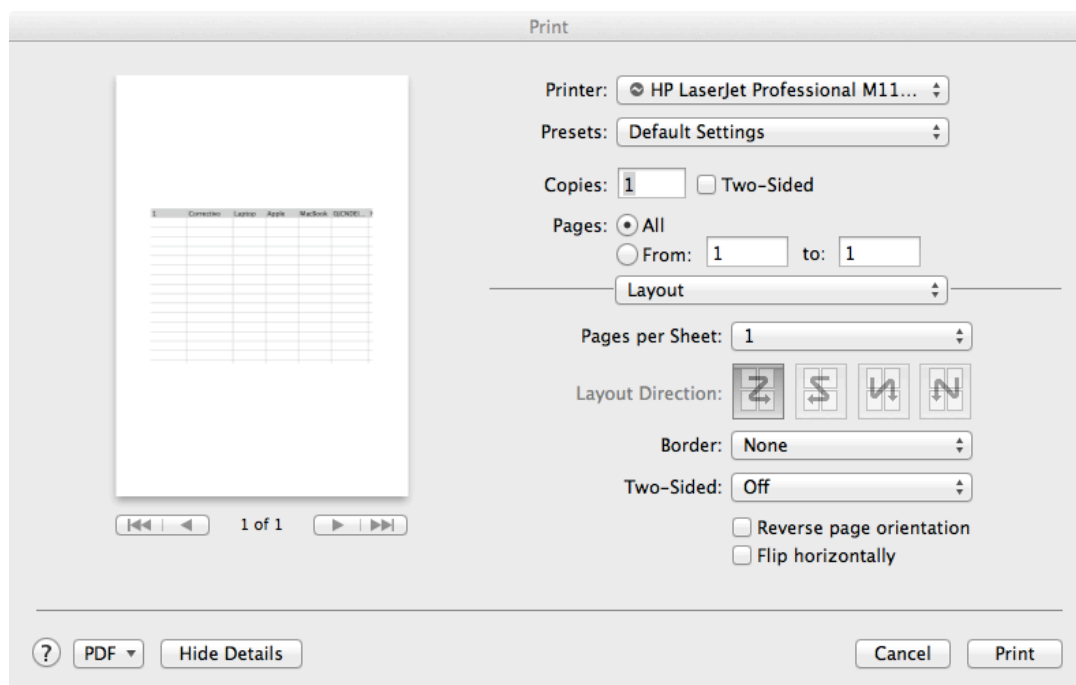


Figura 6.66: Pantalla previa de Impresión (Vista con Detalles).

CAPITULO No. 7:
“IMPLEMENTACION DE LAS
APLICACIONES: *iShelperApp* Y
***iShelperForMac*“**



Figura 7.1: Logotipo de *iShelper_App*.

A continuación, se explica la Implementación de las Aplicaciones “*iShelperApp*” y “*iShelperForMac*”, para eso se utilizó el Programa “*Xcode 4 Versión 4.3.2*”, como se puede ver en la Figura 7.2, y para la aplicación “*iShelperApp*” se utilizó el Programa “*Simulador iOS Versión 5.1*”, como se puede ver en la Figura 7.3:



Figura 7.2: Xcode 4 Versión 4.3.2



Figura 7.3: Simulador iOS Versión 5.1

7.1. iShelperApp:

Ahora, se va a mostrar la Implementación de esta Aplicación mediante las siguientes imágenes y con su respectiva explicación:

Como se había explicado en el Capítulo anterior, esta aplicación se realizó mediante la opción de “Storyboard”, la cual muestra todas las pantallas que hay y la secuencia que tienen para generar los Reportes de Soporte Técnico, como se ve en la Figura 7.4:

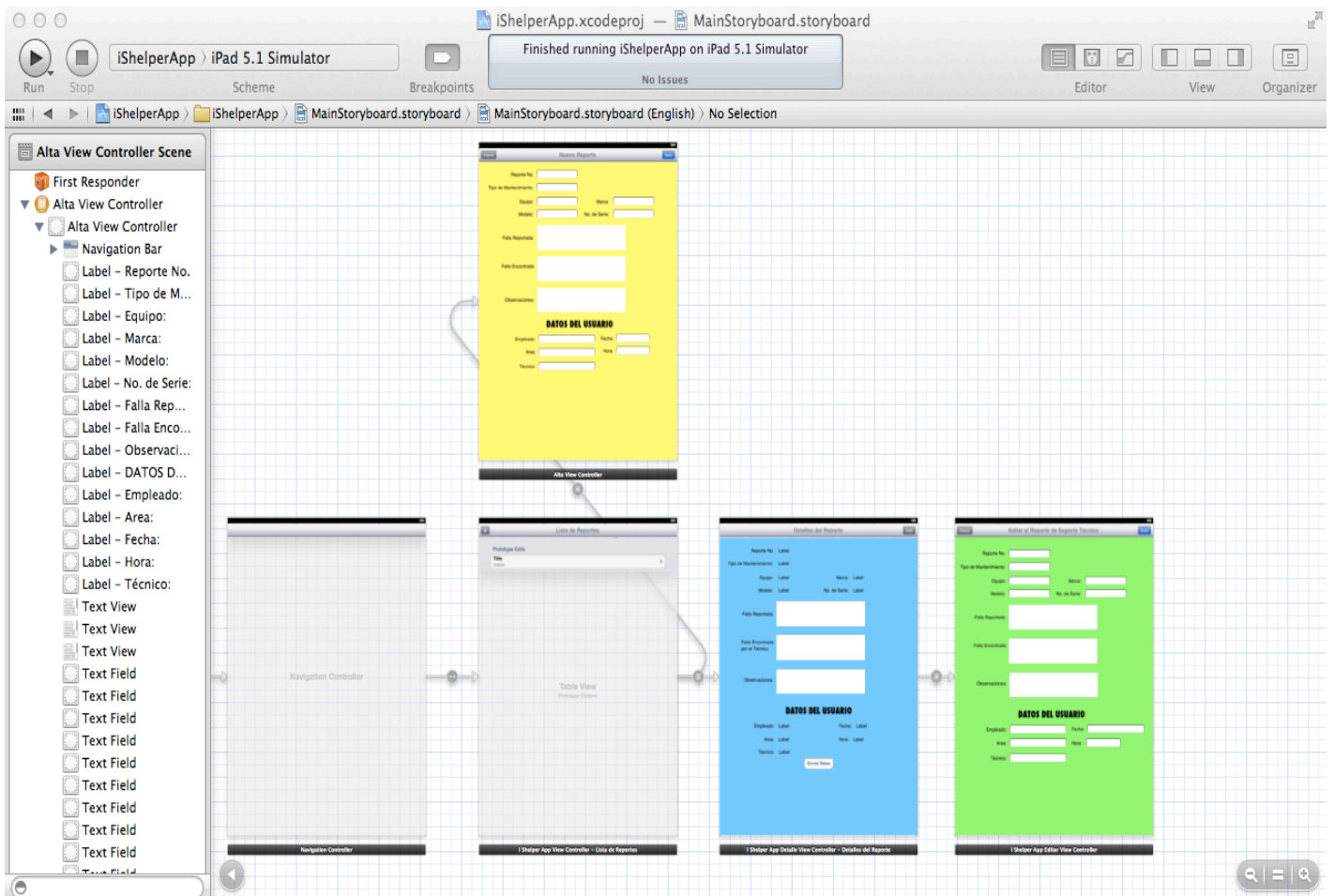


Figura 7.4: Storyboard con todas las pantallas que tiene la Aplicación.

Después, se corre la aplicación el botón “Run”, el cual se puede ver en la Figura 7.5:

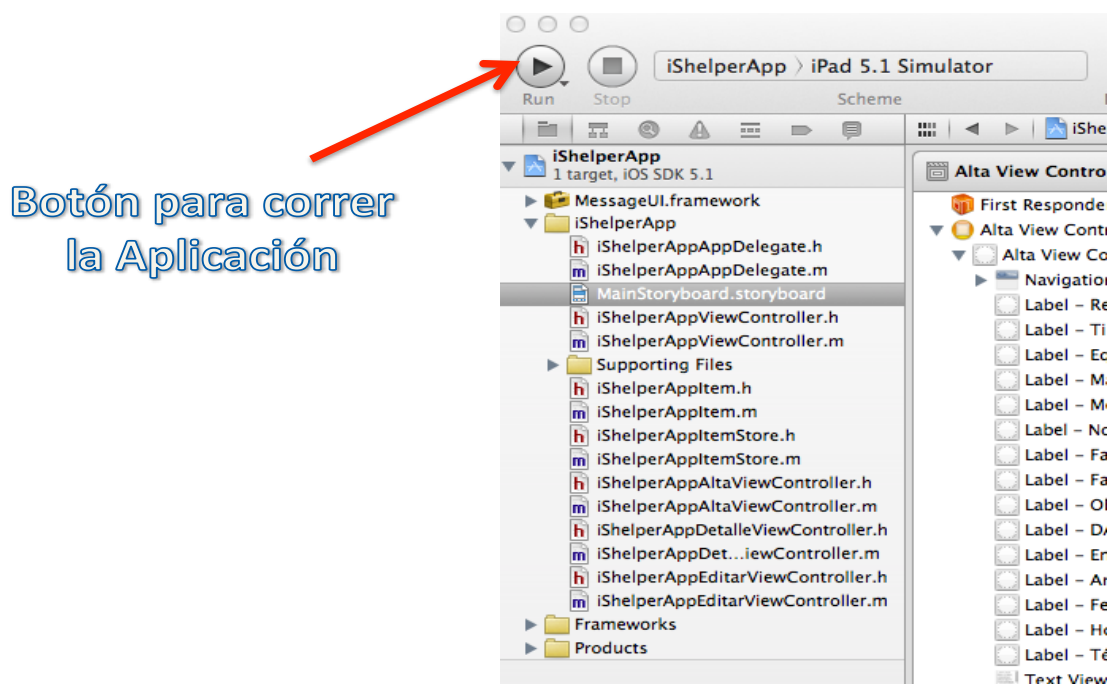


Figura 7.5: Botón *Run* que sirve para correr la Aplicación en el Simulador iOS.

La primer pantalla que se muestra después de correr la aplicación es la que contiene la Tabla de los Reportes que se han generado, la cual aparece vacía por el momento, como se puede observar en la Figura 7.6:

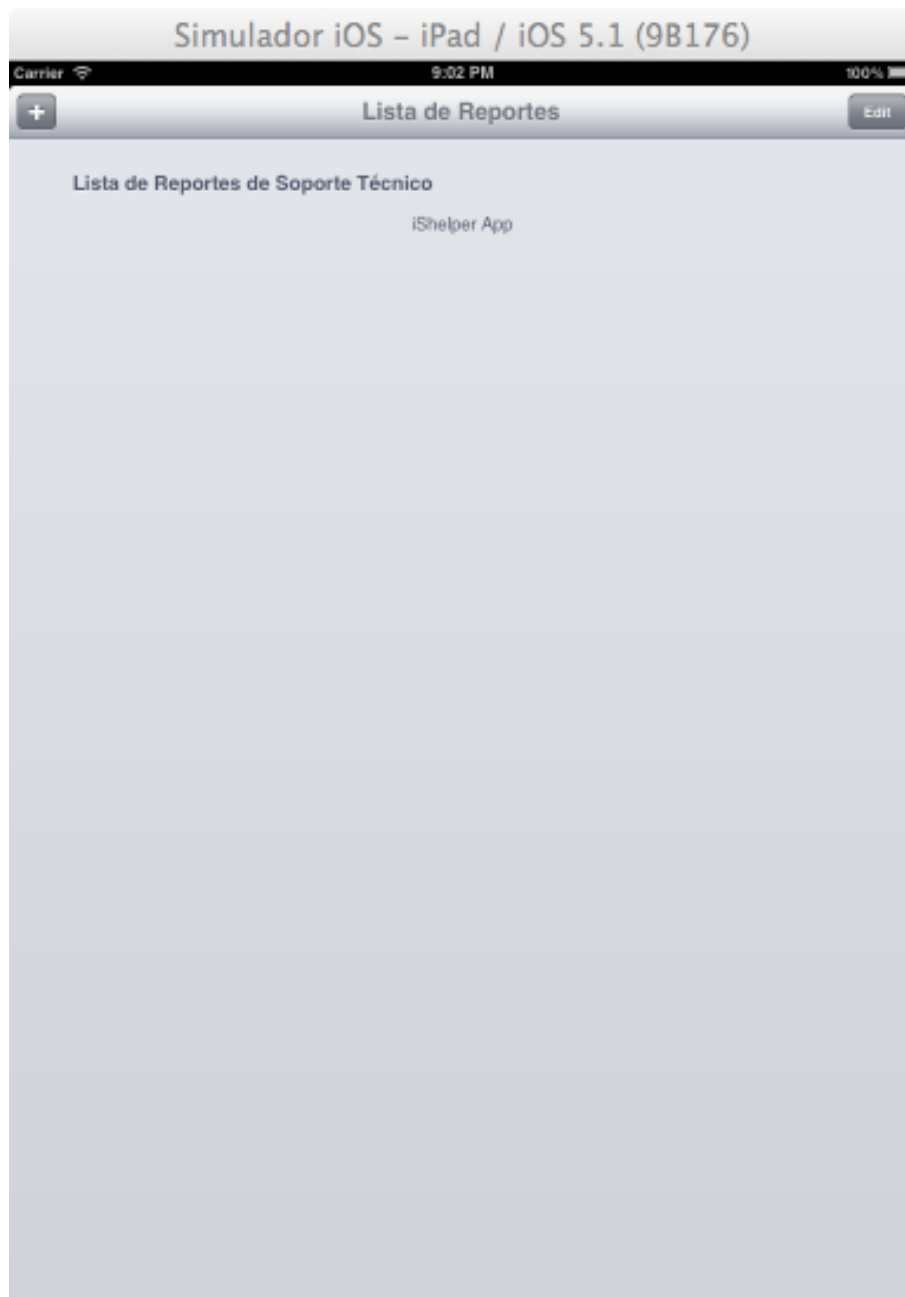


Figura 7.6: Tabla de la Lista de Reportes vacía por el momento.

Esta pantalla cuenta con dos botones, los cuales son el botón "+", que sirve para generar un Reporte nuevo, el cual se va a agregar a la Tabla en forma de Lista, y el botón "Edit", su función es la de eliminar un registro de algún Reporte que se haya realizado previamente, los cuales se muestran en la Figura 7.7:



Figura 7.7: Botón “+” que sirve para generar un reporte nuevo y el Botón *Edit* que sirve para editar la Lista de Reportes.

Después de que se oprime el botón “+”, aparece la pantalla de la Captura de Datos para generar un nuevo Reporte de Soporte Técnico, el cual se muestra en la Figura 7.8:

DATOS DEL USUARIO

Figura 7.8: Pantalla de Captura de Datos para generar un reporte nuevo.

En esta pantalla se puede ver que tiene dos botones, los cuales son el botón “Save”, el cual sirve para guardar los datos que se han introducido en los respectivos campos del formulario, y también está el botón “Cancel”, el cual cancela los datos que se han introducido en los campos del formulario, como se muestra en la Figura 7.9:



Figura 7.9: Botones Save y Cancel de la Pantalla de Captura de Datos.

Cuando se presiona el botón “Cancel”, lo que hace es regresar a la pantalla en donde está la Tabla en forma de lista con los Reportes de Soporte Técnico, la cual no tiene ningún registro, como se muestra en la Figura 7.10:



Figura 7.10: Tabla de la Lista de Reportes vacía al Cancelar la Captura de Datos.

Como se puede ver, se han generado tres Reportes de Soporte Técnico (*uno de Mantenimiento Correctivo y dos de Mantenimiento Preventivo*) en la Tabla de la Lista de Reportes, como se puede ver en la Figura 7.11:



Figura 7.11: Tabla de la Lista de Reportes generadas con los Reportes de Mantenimiento Correctivo, así como del Mantenimiento Preventivo.

A continuación, se muestran en la Figuras 7.12, 7.13 y 7.14, los Reportes de Soporte Técnico que se han generado por el momento:

Simulador iOS – iPad / iOS 5.1 (9B176)

Carrier 1:16 AM 100%

Lista de Reportes Detalles del Reporte Edit

Reporte No. Reporte No. 1

Tipo de Mantenimiento: Mantenimiento Correctivo

Equipo: Laptop Marca: Apple, Inc.

Modelo: MacBook Pro No. de Serie: 864JK873KLMX927U

Falla Reportada: No reconoce la batería.

Falla Encontrada por el Técnico: Aparte de que no reconoce la batería, se activa el ventilador y se calienta demasiado el equipo.

Observaciones: El equipo se encuentra en buen estado.

DATOS DEL USUARIO

Empleado: Alberto Rodriguez Hernandez Fecha: 20 - Septiembre - 2012

Area: Recursos Humanos Hora: 14:45 hrs.

Técnico: Fernando Andres Martinez Fabian

Enviar Datos

Figura 7.12: Reporte No. 1

Simulador iOS - iPad / iOS 5.1 (9B176)

Carrier 1:26 AM 100%

Lista de Reportes Detalles del Reporte Edit

Reporte No. Reporte No. 2

Tipo de Mantenimiento: Mantenimiento Preventivo

Equipo: Impresora Marca: Hewlett-Packard

Modelo: LaserJet 1132 MFP No. de Serie: UEHBD4736NKS

Falla Reportada: El toner que utiliza la impresora se termino.

Falla Encontrada por el Técnico: El toner de la impresora se encuentra vacio en su totalidad.

Observaciones: Hay que cambiar el toner por uno nuevo. La impresora se encuentra en buen estado.

DATOS DEL USUARIO

Empleado: Rodrigo Nava Mejia Fecha: 22 - Septiembre - 2012

Area: Almacen Hora: 17:15 hrs.

Técnico: Fernando Andres Martinez Fabian

Enviar Datos

Figura 7.13: Reporte No. 2

Simulador iOS – iPad / iOS 5.1 (9B176)

Carrier 1:27 AM 100%

Lista de Reportes Detalles del Reporte Edit

Reporte No. Reporte No. 3

Tipo de Mantenimiento: Mantenimiento Preventivo

Equipo: Mouse (Dispositivo Periferico) Marca: Sony

Modelo: JSJD-KND No. de Serie: S01-746HNSU-01

Falla Reportada: No reconoce el cursor en la pantalla.

Falla Encontrada por el Técnico: El mouse se encuentra bastante sucio, por eso no lo reconoce el CPU al momento de utilizarlo.

Observaciones: El Mouse se encuentra en estado deplorable, se sugiere cambiarlo por otro Mouse.

DATOS DEL USUARIO

Empleado: Leticia Garcia Hinojosa Fecha: 24 - Septiembre - 2012

Area: Direccion General Hora: 13:00 hrs.

Técnico: Fernando Andres Martinez Fabian

Enviar Datos

Figura 7.14: Reporte No. 3

Ahora, en la pantalla de la Tabla que contiene la Lista de Reportes que se han generado, cuenta con dos botones que son el botón “+”, el cual ya se explicó anteriormente, por consiguiente se va a explicar la función del botón “*Edit*”, que se muestra en la Figura 7.15:



Figura 7.15: Botones “+” y *Edit* de la Tabla de la Lista de Reportes.

Después de oprimir el botón “*Edit*”, aparecen unos círculos rojos con una raya blanca en medio del lado izquierdo de la lista de Reportes que se han generado, el cual indica que se quiere eliminar un registro de la Lista de Reportes, como se pueden apreciar en las Figuras 7.16 y 7.17:

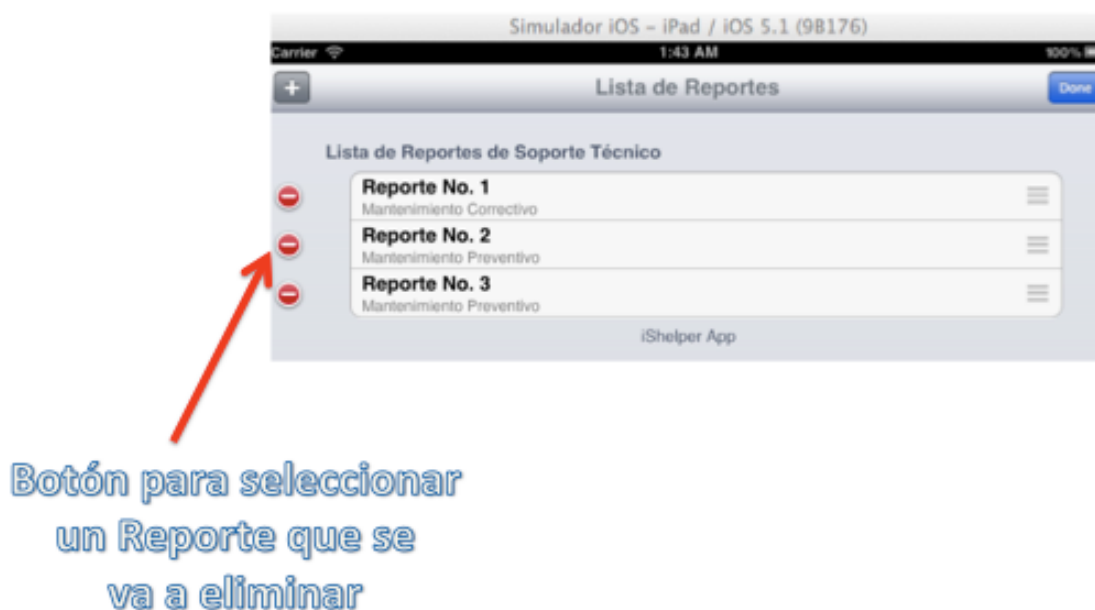


Figura 7.16: Selección de un Reporte que va a ser eliminado.



Botón para eliminar
un Reporte seleccionado

Figura 7.17: Botón *Delete* para eliminar definitivamente un Reporte seleccionado.

Después de presionar el botón “*Delete*”, se elimina el registro seleccionado de la Lista de Reportes, y ahora ya se tiene una nueva pantalla, la cual se muestra en la Figura 7.18:

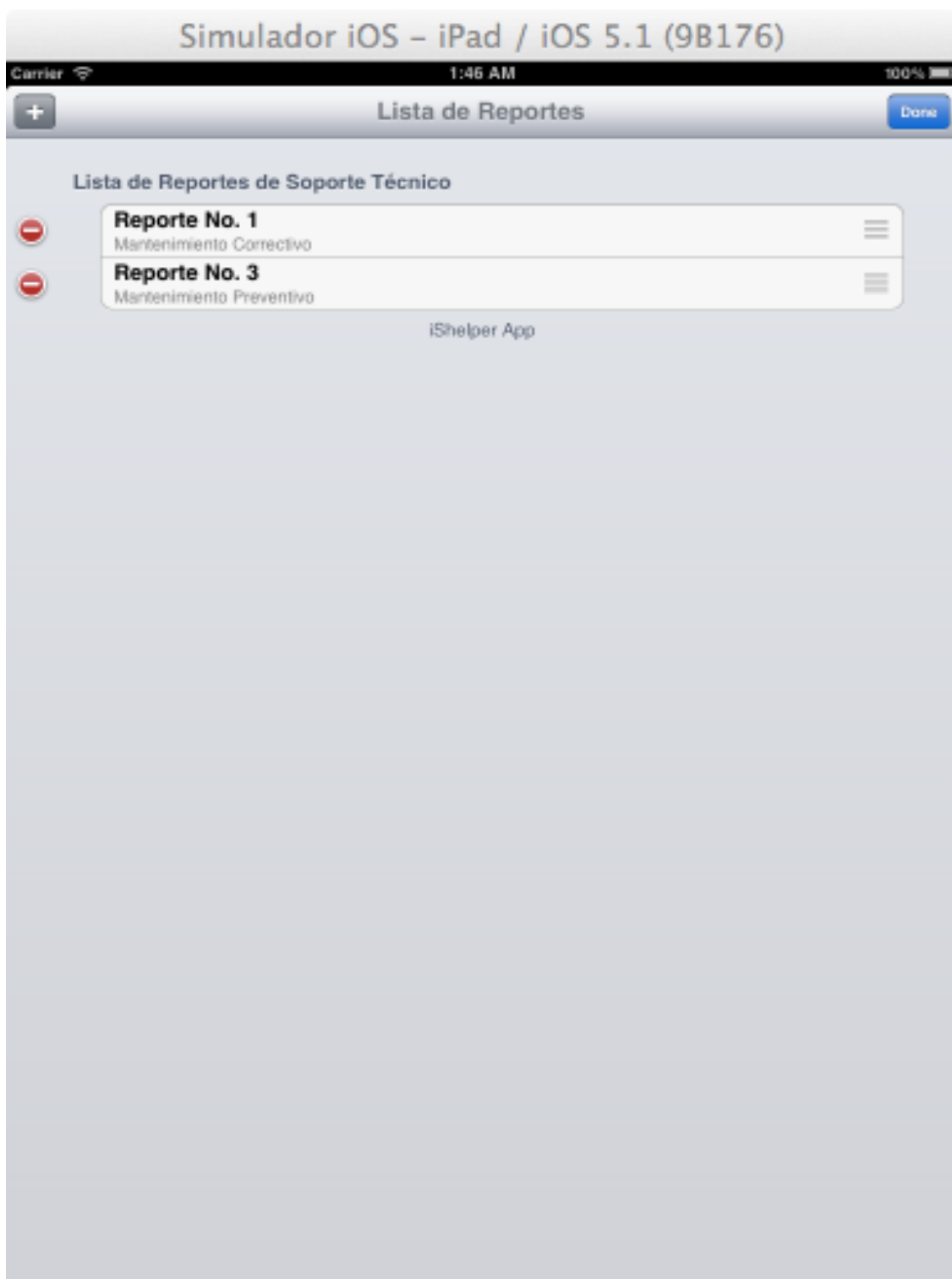


Figura 7.18: Tabla de la Lista de Reportes después de haber sido eliminado de forma definitiva, algún Reporte seleccionado

Ahora, en la pantalla que contiene los Detalles del Reporte que se han generado, tiene un botón de “*Enviar Datos*”, como se puede ver en la Figura 7.19, cuya función es enviar por correo electrónico los datos del Reporte que se han recabado previamente, para eso ya definida la plantilla de correo electrónico, como se puede ver en la Figura 7.20:



Figura 7.19: Botón *Enviar Datos* que sirve para enviar por Correo Electrónico los Datos que han sido recabados previamente.

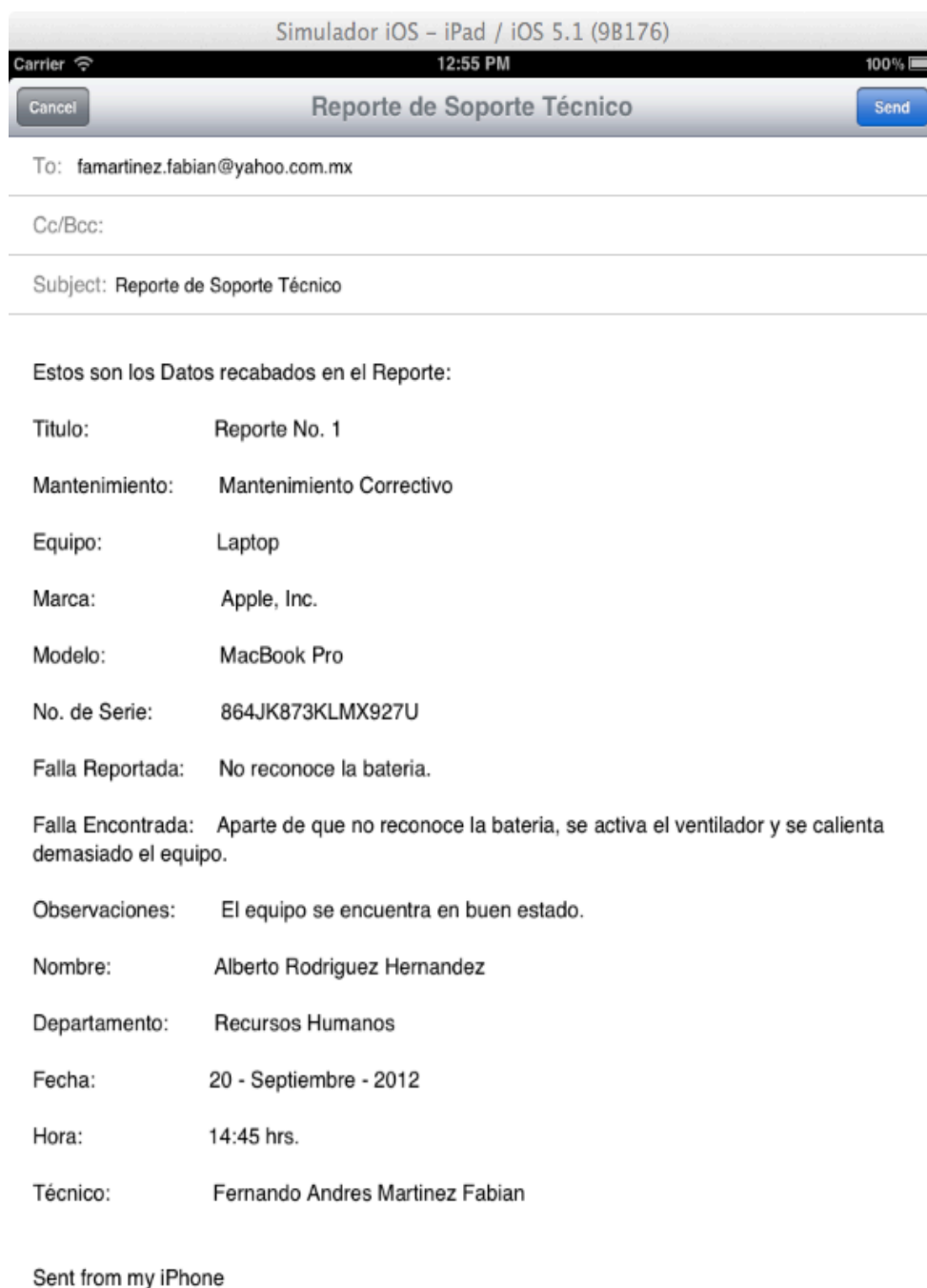


Figura 7.20: Pantalla del Correo Electrónico con todos los Datos capturados previamente.

En la pantalla del correo electrónico, hay dos botones que son el botón “Send”, cuya función es enviar el correo electrónico con los datos capturados del Reporte de Soporte Técnico vía Inalámbrica, y el botón “Cancel”, el cual cancela el envío del correo electrónico, como lo muestra la Figura 7.21:



Figura 7.21: Botón *Send* para enviar el Correo Electrónico y el botón *Cancel* para cancelar el envío del Correo Electrónico.

Ahora, en la pantalla que contiene los Detalles del Reporte, hay un botón que dice “*Edit*”, el cual sirve para modificar los datos que se habían introducido previamente, y después de oprimir el botón “*Edit*”, como se observa en la Figura 7.22, aparece en la pantalla que nos permite modificar los datos introducidos, por si se requiere hacer algún cambio, como se muestra en la Figura 7.23:



Figura 7.22: Botón *Edit* de la Pantalla de *Detalles del Reporte* para modificar los Datos que han sido capturados previamente.

Simulador iOS – iPad / iOS 5.1 (9B176)

Carrier 1:00 PM 100%

Cancelar Editar el Reporte de Soporte Técnico Done

Reporte No. Reporte No. 1

Tipo de Mantenimiento: Mantenimiento Corre...

Equipo: Laptop Marca: Apple, Inc.

Modelo: MacBook Pro No. de Serie: 864JK873KLMX927U

Falla Reportada: No reconoce la batería.

Falla Encontrada: Aparte de que no reconoce la batería, se activa el ventilador y se calienta demasiado el equipo.

Observaciones: El equipo se encuentra en buen estado.

DATOS DEL USUARIO

Empleado: Alberto Rodriguez Hernandez Fecha: 20 - Septiembre - 2012

Area: Recursos Humanos Hora: 14:45 hrs.

Técnico: Fernando Andres Martinez Fab...

Figura 7.23: Pantalla de Edición de los Datos del Reporte que se ha generado.

Esta pantalla tiene dos botones que son el botón “*Done*”, el cual sirve para guardar los datos que han sido modificados en esta pantalla, y el botón “*Cancel*” que sirve para cancelar la modificación de los datos nuevos que se han introducido en esta ocasión, como se muestra en la Figura 7.24:



Figura 7.24: Botón *Done* que sirve para guardar la Modificación de los Datos y el Botón *Cancel* que sirve para Cancelar la Modificación de los Datos.

Por último, se muestra en la Figura 7.25 la pantalla con los Detalles del Reporte, ya sea que hayan sido modificados o no los datos introducidos:

Simulador iOS – iPad / iOS 5.1 (9B176)

Carrier 1:06 PM 100%

Lista de Reportes Detalles del Reporte Edit

Reporte No. Reporte No. 10

Tipo de Mantenimiento: Mantenimiento Correctivo

Equipo: Laptop Marca: Apple

Modelo: MacBook Pro No. de Serie: 864JK873KLMX927U

Falla Reportada: No reconoce la batería.

Falla Encontrada por el Técnico: Aparte de que no reconoce la batería, se activa el ventilador y se calienta demasiado el equipo.

Observaciones: El equipo se encuentra en buen estado.

DATOS DEL USUARIO

Empleado: Alberto Rodriguez Hernandez Fecha: 21 - Septiembre - 2012

Area: Recursos Humanos Hora: 18:45 hrs.

Técnico: Fernando Andres Martinez Fabian

Enviar Datos

Figura 7.25: Pantalla con los *Detalles del Reporte*, ya sea modificados o no los Datos que se ingresaron previamente.

7.2 iShelperForMac:

La función principal de esta Aplicación es la de generar una base de datos con los datos recabados de cada uno de los Reportes generados en el iPad, así como Imprimir la Tabla con los datos recabados.

A continuación, se explica el funcionamiento de la Aplicación junto con sus respectivas imágenes:

Para correr la aplicación, se realiza el mismo procedimiento que en la aplicación anterior, primero se abre el archivo de Xcode, como se muestra en la Figura 7.26, luego se da click en el botón "Run" para correr la aplicación, como se puede apreciar en la Figura 7.27:



Figura 7.26: Pantalla de Bienvenida de Xcode 4.

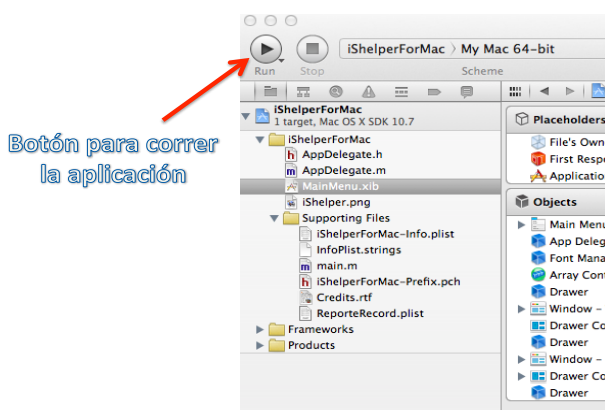


Figura 7.27: Botón Run para correr la Aplicación.

The image shows a software interface for data entry. On the left is a form with the following fields: 'Reporte No.', 'Mantenimiento:', 'Equipo:', 'Marca:', 'Modelo:', 'No. de Serie:', 'Falla Reportada:', 'Falla Encontrada:', 'Observaciones:', 'Empleado:', 'Departamento:', 'Fecha:' (with a date picker set to 26/09/2012), and 'Técnico Responsable:'. On the right side of the form are three buttons: 'Cerrar', 'Agregar Datos', and 'Borrar Datos'. Four colored arrows point from text labels to these buttons and the form area: a green arrow points to 'Cerrar', a blue arrow points to 'Agregar Datos', a pink arrow points to 'Borrar Datos', and an orange arrow points to the form fields. The text labels are: 'Botón para cerrar la ventana', 'Botón para agregar Datos en la Tabla', 'Botón para borrar los datos de un renglón de la Tabla', and 'Formulario para introducir los datos en los campos que sean requeridos'.

Figura 7.29: Formulario para la Captura de Datos para generar la Base de Datos con los Botones *Agregar Datos*, *Borrar Datos* y *Cerrar*.

Por último, se explica cuál es la función del botón "*Imprimir Tabla*", cuyo funcionamiento es la de imprimir la tabla que contiene los datos que se han ingresado mediante el formulario, como se muestra en la Figura 7.30:

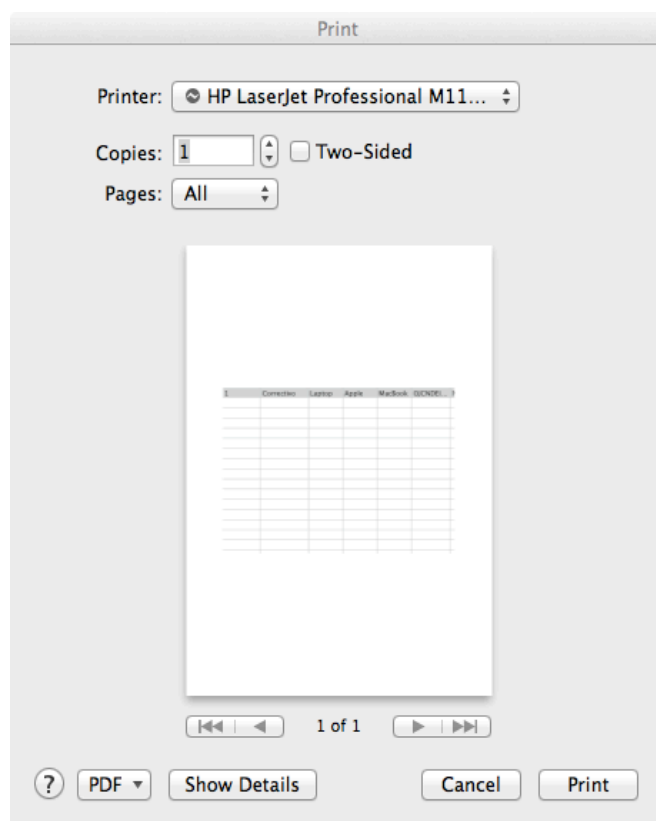


Figura 7.31: Pantalla previa para Imprimir (Vista sin Detalles).

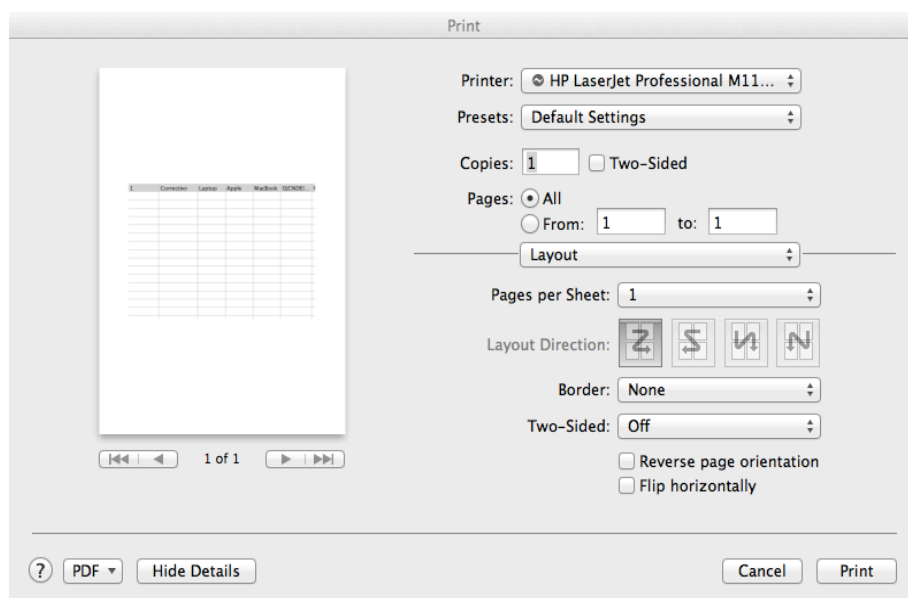


Figura 7.32: Pantalla previa para Imprimir (Vista con Detalles).

CONCLUSIONES Y RECOMENDACIONES

8.1 Conclusiones:

El desarrollo de aplicaciones y sistemas para dispositivos móviles, es cada vez más importante en el ámbito laboral, y es por eso que este trabajo de titulación se propone como una aportación de muchas investigaciones posteriores para los estudiantes de la Facultad de Contaduría y Administración sobre el desarrollo para dispositivos móviles, así como las herramientas que se utilizaron, como son el programa Xcode, el lenguaje de programación Objective-C, bases de datos, etc.

Este trabajo de titulación presenta los resultados de una investigación sobre los lenguajes, tecnologías y herramientas que fueron utilizados para la programación de los dispositivos móviles, así como el conocimiento de cómo funciona el área de Soporte Técnico dentro de las Empresas, en particular se presentaron las tecnologías que actualmente ofrece Apple, Inc., como es el iPhone, el iPod Touch, el iPad, la iMac, la MacBook, la MacBook Pro y la MacBook Air para el desarrollo sobre estos dispositivos, así como también el uso de las bases de datos mediante el programa SQLite, todo esto con el objetivo de mostrar la forma de cómo cualquier estudiante, puede realizar un Sistema Distribuido para un Dispositivo Móvil.

Se realizó la aplicación **“iShelperApp”**, la cual consiste en un sistema de generación de reportes de soporte técnico que está destinada a que funcione en el iPad, para que la información se envíe vía inalámbrica hacia una computadora localizada en el área de Soporte Técnico de la Empresa, para que se pueda llevar a cabo la toma de decisiones de que actividad se va a proceder a realizar para el Mantenimiento Correctivo del Equipo Informático, así como también el Mantenimiento Preventivo del mismo.

También se realizó la Aplicación **“iShelperForMac”**, la cual consiste en una base de datos para las computadoras iMac, MacBook, MacBook Pro y MacBook Air, ya que se introducen los datos recabados desde la Aplicación **“iShelperApp”** en los campos determinados para que aparezcan dentro de la Tabla Principal, para que posteriormente se le pueda dar un uso adecuado a esa información.

Finalmente, se puede informar que esta investigación y proyecto conllevará a la realización de sistemas más complejos, los cuales se podrán utilizar ampliamente por las empresas y las personas, además de que también se considerará el desarrollo de aplicaciones para los dispositivos móviles como complemento para cualquier otro Sistema creado para computadoras de escritorio, logrando así Sistemas bastante interesantes, más útiles y que le den tanto al usuario como al desarrollador satisfacción por el trabajo realizado.

8.2 Recomendaciones:

Se ha desarrollado este trabajo de titulación, proyectándolo a un futuro cercano en la investigación y el desarrollo de las tecnologías, Sistemas y aplicaciones para dispositivos móviles, especialmente para los estudiantes de la Licenciatura en Informática de la Facultad de Contaduría y Administración de la UNAM, para que obtengan un nivel de conocimientos mayor en el área de desarrollo y las oportunidades de trabajo aumenten, así como también se pueda tomar el desafío de crear una empresa de desarrollo de aplicaciones, en donde se consideren este tipo de dispositivos, los cuales van en aumento y el desarrollo enfocado a dispositivos móviles, es poco, debido a que son pocas las personas que se dedican a esto.

En cuanto a la aplicación que este trabajo de titulación presenta, se puede mejorar bastante, tanto para que se lleve a cabo de una mejor forma, el intercambio de información entre un dispositivo móvil y la Mac, así como también para añadir un mayor interés al usuario en utilizarlas.

A continuación se mencionan algunas de las mejoras que se van a realizar a mediano plazo:

- Ampliación del diseño y capacidades del Sistema.
- Definir las métricas que aumenten la seguridad del Sistema.
- Optimización del tiempo de consulta y respuestas.

ANEXO A: “DISPOSITIVOS DE APPLE, INC.”

A.1 iPhone:



Figura A.1: iPhone

Es la combinación de tres productos: un revolucionario teléfono móvil, un iPod de pantalla completa con controles táctiles, y un revolucionario dispositivo de comunicación que a través de Internet con email, al mismo nivel que una computadora, navegación web, búsquedas y mapas; todo ello que se encuentra integrado en un pequeño y ligero dispositivo portátil de mano; además de que tiene una nueva interfaz de usuario basada en una pantalla táctil “multi-touch”, y en un pionero e innovador software que permite al usuario controlarlo sólo con sus dedos.

El iPhone abre una nueva era de potencia y sofisticación en el software nunca antes visto en un dispositivo móvil, ya que redefine completamente lo que los usuarios pueden hacer con sus teléfonos móviles.

El iPhone como revolucionario Teléfono Móvil:

El iPhone es un nuevo y revolucionario teléfono móvil, ya que permite a los usuarios hacer llamadas con sólo señalar un nombre o número, también puede sincronizar todos los contactos con una PC, iMac, MacBook o servicio de Internet como Yahoo!, de tal manera que se pueda llevar con uno mismo la lista de contactos completa y actualizada. Además se pueden construir más fácil la lista de favoritos para las llamadas más frecuentes, así como también combinar fácilmente las llamadas para crear conferencias grupales.

El iPhone incluye una aplicación SMS con un teclado “QWERTY”, para enviar y recibir mensajes SMS en múltiples sesiones. Cuando los usuarios necesitan teclear, el iPhone les presenta en la pantalla un elegante teclado táctil con capacidad productiva que corrige errores, mucho más fácil y más eficiente de usar, a comparación de los pequeños teclados de plástico de varios Smartphones, por ejemplo el BlackBerry; además de que incluye una aplicación calendario, el cual permite sincronizar automáticamente los calendarios con la PC, iMac, MacBook.

El iPhone es un teléfono GSM cuatribanda que incorpora las tecnologías inalámbricas EDGE y Wi-Fi para redes de datos.

El iPhone es un iPod de Pantalla Completa:

El iPhone es un iPod de pantalla completa, con toda la superficie plana, ya que cuenta con controles táctiles que permiten a los usuarios “tocar” la música, desplazándose fácilmente a través de las listas de canciones, artistas, álbumes y las listas de reproducción con sólo mover un dedo. Las carátulas de los álbumes se muestran de manera espectacular en la gran y brillante pantalla del iPhone.

El iPhone permite al usuario disfrutar de todo el contenido de su iPod, incluyendo música, audiolibros, podcasts de audio y video, videos musicales, programas de TV y películas. El iPhone sincroniza su contenido con la Biblioteca iTunes del usuario en su PC, iMac o MacBook, y puede reproducir cualquier contenido de música o de video comprado en la Tienda iTunes.

El iPhone como revolucionario Dispositivo de Comunicación a través de Internet:

El iPhone incluye un avanzado cliente de email HTML, que extrae el email en segundo plano de la mayoría de los servicios de correo electrónico POP3 o IMAP, y visualiza fotos y gráficos junto al texto. El iPhone es totalmente multitarea, de tal manera que se puede estar leyendo una página web, mientras que se esté descargando el email en segundo plano.

Además de que incorpora el más avanzado navegador web presente en un dispositivo móvil; se trata de una versión del navegador web Safari de Apple. Los usuarios pueden ver cualquier página web, tal como ha sido diseñada para ser vista, y se puede hacer zoom de forma fácil para ampliar cualquier zona, con el solo hecho de tocar la pantalla táctil multi-touch del iPhone con el dedo. Los usuarios pueden explorar la web desde cualquier parte con la Tecnología Wi-Fi o EDGE, además de que pueden sincronizar de forma automática sus favoritos con la PC, iMac o MacBook. El navegador web Safari del iPhone integra también “Google Search” y “Yahoo! Search”, de tal modo que los usuarios pueden buscar la información de manera instantánea en su iPhone, como si lo hicieran en una computadora.

A.2 iPod:



Figura A.2: iPod

Es una línea de reproductores de audio digital portátiles, diseñados y comercializados por Apple, Inc. Fue presentado por primera vez el 23 de octubre de 2001.

Generalidades:

La línea de productos consiste en los Modelos iPod Classic, iPod Touch, iPod Nano y el iPod Shuffle. El iPhone puede funcionar como un iPod, pero se le trata como un producto por separado. Algunos Modelos anteriores, y que ya se encuentran fuera del catálogo de la Familia iPod son el iPod Mini y el iPod Photo (esto sucedió cuando se integró la línea de iPod Classic), aunque en 2007 se descubrió que había grandes errores de producción: 1 de cada 100 productos de Apple, Inc., tenía el número de serie repetido. Esto fue una falla de las máquinas de grabado, pero existe otro número en el condensador (que es la memoria en donde se almacena toda la información), el cual fue el que permitió la distinción. Fue la única manera de ver el número de serie real.

El iPod Classic almacena la multimedia en un disco duro interno, mientras que el resto de los Modelos hace uso de una memoria flash para permitir su menor tamaño (el iPod Mini utilizaba un disco duro Microdrive). Como ocurre con varios reproductores de música digital, los iPod, a excepción del iPod Touch, también puede servir como dispositivos de almacenamiento de datos externos. La capacidad de almacenamiento varía según el Modelo (el de mayor capacidad es el iPod Classic que es de 160 GB).

El software llamado "Apple iTunes", se utiliza para transferir música al iPod desde una computadora, la cual utiliza ciertas aplicaciones Apple para Macintosh y Microsoft Windows. Para los usuarios que eligen no utilizar el software de Apple o en cuyos equipos no se puede ejecutar el programa "iTunes", existen varias alternativas de código abierto, aunque no es posible para ningún fabricante sin licencia garantizar un buen funcionamiento, ya que Apple no brinda las especificaciones con respecto a la interfaz de software del iPod, solamente soporta oficialmente "iTunes", como su única interfaz.

Desde el programa "iTunes" se pueden transferir fotos, videos, juegos, información de contacto, correo electrónico, favoritos web, así como también calendarios para los Modelos de iPod que soportan estas características. Apple se centró en el desarrollo de su línea de iPod, en la interfaz de usuario única y su facilidad de uso, en lugar de la capacidad técnica. A partir de Septiembre de 2007, se han vendido en todo el mundo alrededor de 150 millones de iPod, convirtiéndose así en el reproductor de música digital más vendido.

Software:

El iPod puede reproducir archivos MP3, WAV, AAC/M4A, AIFF y Apple Lossless. El iPod de 5ª Generación también es capaz de reproducir archivos de video en formatos H.264 y MPEG-4.

Apple diseñó el iPod para funcionar principalmente en combinación con su programa iTunes, ya que con este programa se puede sincronizar automáticamente la música de la Biblioteca del usuario con la del iPod cada vez que se conecta con la computadora. Las canciones, fotos y listas de reproducción nuevas se copian automáticamente al iPod y las canciones borradas son eliminadas también del iPod. Esto funciona también en sentido inverso, si se valora una canción en el iPod, la calificación se copiará a la Biblioteca de iTunes. Además, si se escuchan audiolibros, la posición de éstos también se sincronizará de forma automática. iTunes da la opción de sincronización manual, además de la función de sincronización automática. El iPod de 5ª Generación, el iPod Classic y la última generación de reproductores iPod (excepto el Shuffle), poseen la capacidad de descargar juegos de diversa temática desde el “iTunes Store”.

A.3 iPod Touch:



Figura A.3: iPod Touch

Es el primer iPod que posee una interfaz táctil y comparte muchas de las características de vanguardia introducidas por primera vez con el iPhone. El iPod Touch es mucho más que un reproductor de música, es un iPod capaz de reproducir audio, video y en el que se pueden visualizar fotos, pero también es un navegador de Internet, posee agenda, calendario y permite descargar videos de YouTube, además que es el primer iPod en el que se puede navegar, comprar y descargar canciones desde la “iTunes Store”.

Introducción:

El iPod Touch es al mismo tiempo un iPod y un navegador de Internet, que posee una avanzada interfaz de usuario, que es sensible al tacto que utiliza tecnología multi-touch. Es hasta el momento, el iPod más avanzado y es el primer reproductor iPod que se aleja drásticamente del formato original del iPod con una pantalla en la parte superior y una rueda de control bajo ella. El Touch es más parecido a un iPhone, pero sin teléfono que a un iPod. El iPod Touch se ofrece en tres versiones, una de 8 GB, otra de 32 GB y una más de 64 GB.

Exterior y Diseño:

El iPod Touch tiene forma rectangular, es más alto que ancho y tiene realmente muy poco espesor, ya que se puede confundir con un iPhone, porque tiene el mismo formato y casi las mismas dimensiones; además el iPhone Touch es más delgado que el iPhone y no tiene auricular, ni altavoces.

El frente del iPod Touch está denominado por la gran pantalla color sensible al tacto. Es una pantalla de 3.5 pulgadas con formato panorámico, ya que es ideal para ver películas o videos. Bajo esta pantalla se encuentra el único botón del frente del iPod, y mediante el cual se vuelve al Menú de Inicio desde cualquier pantalla en la que esté abierta en ese momento.

La pantalla está recubierta por una lámina de vidrio, y alrededor de la pantalla hay un borde de color negro, que también está cubierto por la lámina de vidrio. Los bordes del iPod son redondeados y de color negro.

La parte posterior del iPod Touch es de acero inoxidable pulido y con bordes redondeados como la de los iPod originales. En la esquina superior izquierda, hay una pequeña parte que es de plástico negro, y bajo ella se encuentra la antena para la red Wi-Fi. En el centro de la cara posterior del iPod Touch se encuentra el logo de Apple y bajo él, la palabra "iPod". Un poco más abajo está grabada la capacidad de este Modelo y bajo la capacidad, se encuentra el número de serie del iPod, el Modelo y otra información.

En la parte superior del iPod Touch, se encuentra el botón que se utiliza para encender, apagar y suspender al iPod. Mirando el iPod de frente, este botón está ubicado a la izquierda, a diferencia del iPhone, en él que este botón está ubicado a la derecha.

En la base del iPod, se encuentra el conector dock en el centro, y a su derecha la salida de los auriculares.

Algo que llama la atención bastante del aspecto exterior es lo delgado que es, sólo tiene 8 mm de espesor, tres menor que el iPhone.

Sensores del iPod Touch:

El iPod Touch hereda dos de los sensores del iPhone. Uno es el sensor de luz, el cual se utiliza para ajustar el brillo de la pantalla del iPod Touch dependiendo de las condiciones de luminosidad del exterior. De esta forma, el iPod utiliza de forma más eficiente la energía de la batería y hace que su pantalla sea más fácil de leer en exteriores.

El otro de los sensores, es un sensor de orientación que detecta si el iPod Touch está en posición vertical horizontal, y esa información es utilizada en algunas aplicaciones para ajustar la ventana.

A.4 iPad:



Figura A.4: iPad

Es un dispositivo electrónico tipo “Tablet”, desarrollado por Apple, Inc. La primera generación fue anunciada el 27 de enero de 2010, mientras que la segunda generación surgió el 2 de marzo de 2011. Se sitúa en una categoría entre un “teléfono inteligente” y una computadora portátil, enfocado más al acceso, que a la creación de aplicaciones y temas.

La tercera versión del dispositivo, que pasó a llamarse de nuevo iPad, fue presentada el 7 de marzo de 2012.

El iPad mini es el nuevo dispositivo de Apple, Inc., fue lanzado junto con el iPad de cuarta generación el 23 de octubre de 2012 en Estados Unidos y otros países de Europa.

Las funciones son similares que al resto de los dispositivos móviles de Apple, Inc., como es el caso del iPhone o el iPod Touch, aunque la pantalla es más grande y su hardware es más potente. Funciona a través de una Interfaz Natural de Usuario, sobre una versión adaptada del Sistema Operativo iOS. Esta interfaz de usuario está rediseñada para aprovechar el mayor tamaño del dispositivo y la capacidad de utilizar software para la lectura de libros electrónicos y periódicos, navegación web y correo electrónico, además de permitir el acceso al usuario a otras actividades de entretenimiento como películas, música y videojuegos.

Posee una pantalla con retroalimentación LED y capacidades multi-táctiles de 9.7 pulgadas, de 16 a 64 GB de espacio en memoria flash, Bluetooth, y un puerto de conexión periférica de 30 pines que permite la sincronización con el software iTunes, además de proporcionar conexión para diversos accesorios. Existen dos Modelos: uno con conectividad a redes inalámbricas Wi-Fi 802.11n y otro con capacidades adicionales de GPS y soporte a redes 3G (puede conectarse a las redes de telefonía celular HSDPA). Ambos Modelos pueden ser adquiridos en tres capacidades de almacenamiento distintas, además de que se pueden adquirir en 2 colores diferentes: en blanco y en negro, ambas versiones están disponibles con 16 GB, 32 GB, 64 GB y 128 GB.

Características:

Hardware:

Sensores:

- Acelerómetro
- Sensor de luz
- Brújula digital
- GPS Asistido (Modelo Wi-Fi + 3G)
- Giroscopio (iPad 2 y el nuevo iPad)

Entrada y Salida:

- Transferencia de datos con tarjeta SIM (Modelo 3G)
- Conexión inalámbrica Wi-Fi
- Toma de auriculares estéreo de 3.5 mm
- Altavoz integrado
- Micrófono
- Puerto conector de 30 pines para colocar hardware externo (Kit iPad Camera Connection, Base iPad Dock, teclado físico)

Controles:

- Pantalla multi-táctil
- Botón de reposo/activación y bloqueo
- Botón de bloqueo de rotación de pantalla o silencio
- Control de volumen
- Botón "Inicio"
- Conector de audífonos
- Conector del USB
- Cable de Apple

Software:

El iPad tiene el siguiente software:

- **App Store:** Software que permite adquirir aplicaciones.
- **Safari:** navegador de Internet con soporte para HTML 5.
- **Mail:** para gestionar correos electrónicos con Microsoft Exchange, MobileMe, Gmail, Yahoo! Mail y AOL.
- **YouTube:** navegador y visualizador de YouTube.
- **iPod:** función equivalente a su análogo reproductor multimedia iPod.
- **iTunes Store:** para comprar música, videos, descargar Podcast y visualizar contenidos.
- **Mapas:** Para visualizar mapas y establecer rutas, utiliza la tecnología de Google Maps, incluido el servicio de Street View.

- **Calendario:** Un calendario para gestionar citas y fechas especiales.
- **Contactos:** Para gestionar los contactos.
- **FaceTime** (Únicamente iPad 2 y el nuevo iPad).
- **Photo Booth** (Únicamente iPad 2 y el nuevo iPad).

A.5 iMac:



Figura A.5: iMac

Es una serie de computadoras todo-en-uno diseñados y fabricados por Apple, Inc. Está orientado al mercado doméstico y todos sus Modelos se caracterizan por integrar el CPU y el monitor en un único aparato. Desde la presentación del primer Modelo en 1998, el iMac ha evolucionado a través de cinco formas distintas, todos firmados por Jonathan Ive, quien es un famoso diseñador industrial.

Esta computadora tiene una popularidad muy alta, debido a que su estética se diferencia del resto de las computadoras de escritorio del Mercado, unas campañas de Mercadotecnia bastante exitosas y una publicidad por emplazamiento muy agresiva en películas y series de producción norteamericana y el resto del mundo.

En 1998, Steve Jobs presentó el primer iMac y, desde entonces, esta pequeña pero importante letra precedería el nombre de sus productos más famosos, ya que representa, entre muchas cosas, palabras clave como son: **internet, individual, instruir, informar e inspirar.**

Modelos de iMac:

- iMac G3
- iMac G4
- iMac G5

iMac Intel:

- iMac Core Duo
- iMac Core 2 Duo/Quad core
- iMac Core 2 Duo (segunda generación)
- iMac Core iX (i3, i5 e i7) (primera generación de 21.5" / segunda generación de 27")

A.6 MacBook:



Figura A.6: MacBook

Es una Familia de computadoras portátiles desarrollados por Apple, y que tienen como Mercado Objetivo los usuarios básicos, de hogar y las pequeñas empresas. Fueron lanzados en Mayo de 2006, reemplazando al iBook y al PowerBook 12" como parte de la transición de Apple hacia la tecnología de Intel.

Llegó a convertirse en el equipo portátil Macintosh más vendido en la historia de la compañía. De acuerdo con el NPD Group (Firma de Investigación de Mercado) en Octubre de 2008, el Modelo Gama Media era la computadora portátil más vendida en tiendas minoristas en Estados Unidos en los 5 meses anteriores.

Han existido tres diferentes diseños del MacBook: el original (el cual tenía una carcasa de policarbonato similar a la del iBook G4). El segundo tipo, introducido en Octubre de 2008 junto con el MacBook Pro 15", tenía una carcasa de aluminio en formato unibody. Este Modelo fue promovido a la MacBook Pro en la Apple Worldwide Conference en junio de 2009. El tercer diseño, introducido en Octubre de 2009, reemplazó la carcasa original con una de policarbonato en formato unibody.

Finalmente con la llegada del Mac OS X Lion el 19 de Junio de 2011, Apple decidió descontinuar el MacBook Blanco. Esto hizo que la MacBook Air de 11" fuera la computadora portátil más barata de Apple, haciendo que su sucesor fuera el MacBook Air.

A.7 MacBook Pro:



Figura A.7: MacBook Pro

Es una línea de computadoras portátiles de alto rendimiento de Apple, Inc., que tiene como Mercado Objetivo los usuarios profesionales. Es el sucesor del Modelo PowerBook G4. Es la primera línea de productos de la Empresa en incluir procesadores Intel, y fue presentada en la Feria Tecnología MacWorld 2006, el 10 de enero de 2006.

Está basado en los procesadores Intel Core 2 Duo (de hasta 2.66 GHz), procesadores Intel Core i5 e Intel Core i7, incluye una videocámara “iSight”, ahora llamada “FaceTime HD”, teclado retroiluminado, sensor de iluminación (el cual controla el brillo de la pantalla y el teclado), la pantalla retroiluminada por LED y además tiene nuevas funciones como el conector MagSafe. Todas sus variantes incluyen sensores de caída libre (los cuales retiran la aguja de escritura del disco duro en caso de que el equipo se caiga). Para Octubre de 2011, el Sistema Operativo incluido es el Mac OS X v10.7 Lion.

Los MacBook posteriores a Junio de 2009 hacen uso de tecnología amigable con el Medio Ambiente: cristal sin arsénico, ausencia de retardantes de llama bromados, pantalla sin mercurio retroiluminada por LED, cables internos sin PVC, carcasa de aluminio y cristal reciclable en su mayor parte, al igual que se ha reducido el embalaje. Están a conformidad con la norma Energy Star (versión 5.0), y obtuvieron la Certificación Gold de EPEAT.

ANEXO B: “PROGRAMAS DE APPLE, INC.”

B.1 Mac OS X 10.8 (Mountain Lion):



Figura B.1: Mac OS X Versión 10.8 (Mountain Lion).

OS X Mountain Lion (versión 10.8) es la novena versión de OS X, el sistema operativo de Apple, Inc., para sus computadoras de escritorio, portátiles y servidores. *Mountain Lion* fue anunciado el 16 de febrero de 2012 y fue puesto a la venta el 25 de julio de 2012 a través de la Mac App Store. Incorpora aún más características del sistema iOS, alguna de las cuales ya han sido introducidas en *Lion*. Diversas aplicaciones fueron renombradas o modificadas para lograr una correspondencia con sus contrapartes de iOS.

Características:

Mountain Lion trae más de 100 nuevas mejoras o nuevas características:

- Mayor integración con *iCloud*, en *Pages*, *Numbers*, *Keynote* y *Vista Previa*.
- Sincronización automática de documentos de la suite *iWork* a través de *iCloud*.
- *Messages*, un cliente multiprotocolo de mensajería instantánea y envío de mensajes de texto. Reemplazará a *iChat* y soportará el servicio *iMessage*, disponible en *iOS*.
- *Recordatorios*, ahora separados de *iCal*.
- *Notas*, ahora separadas de *Mail*.
- Una versión de escritorio del Centro de Notificaciones de iOS, al estilo de *Growl*.
- *AirPlay Mirroring*, que permitirá enviar la imagen de la computadora, a través de AirPlay, hacia el Apple TV.
- *Gatekeeper*, una nueva característica para prevenir o limitar instalar aplicaciones de terceros, y eligiendo así instalar aplicaciones solo de la Mac App Store o de otras vías.
- Integración con Twitter y Facebook en la versión 10.8.2

- *Game Center*, un servicio de juego de iOS que permite compartir logros y puntuaciones con las personas que juegan al mismo tiempo.

B.2 OS X Server:



Figura B.2: OS X Server.

OS X Server, anteriormente conocido como *Mac OS X Server*, es un servidor Unix se vende por separado del Sistema Operativo de Apple, Inc., arquitectónicamente idéntica a su contraparte de escritorio OS X –con programas de servidor adicionales, de gestión y herramientas de administración-. Desde la versión 10.7 (*Lion*), *Mac OS X* y *Mac OS X Server* se combinaron en un comunicado de marca como OS X. Un “servidor” separado del Sistema Operativo ya no se vende, y las especificaciones del servidor de aplicaciones de servidor y grupo de trabajo, las herramientas de gestión y administración de software de *Mac OS X Server* se ofrece ahora como servidor de aplicaciones, un paquete add-on para OS X se vende a través de la Mac App Store, junto con las herramientas de administración del servidor versión 10.7 (*Server Admin Tools 10.7*), desde el sitio web de soporte de Apple.

Estas herramientas simplifican el acceso a los servicios de red clave, entre ellos un agente de transferencia de correo, AFB y servidores SMB, un servidor LDAP, un servidor de nombres de dominio y otros. También se incluyen (particularmente en las versiones posteriores) numerosos servicios adicionales y las herramientas para su gestión, como es el caso del servidor web, servidor wiki, servidor de chat, servidor de calendario, entre otros.

Mac OS X 10.7 (Lion Server):

Lion Server viene con licencias de cliente ilimitadas al igual que *Snow Leopard Server*. Al dar a conocer la vista previa para desarrolladores de *Mac OS X Lion* en Febrero de 2011. Apple, Inc., indicó que a partir de *Lion*, *Mac OS X Server* se incluye con el Sistema Operativo y no se comercializa como un producto separado.

Lion Server incluye nuevas versiones de *iCal Server*, *Servidor Wiki* y el *Servidor de Correo*. Más significativamente, *Lion Server* puede ser utilizado para la gestión de los dispositivos móviles que tengan el *iOS*.

OS X 10.8 (Mountain Lion Server):

No hay una edición de servidor separada de *Mountain Lion*, así como también no había ninguna edición separada de *Lion Server*.

Mountain Lion Server, como el *Lion Server*, viene con licencias de cliente ilimitadas, y una vez comprado se puede ejecutar en un número ilimitado de máquinas.

B.3 iWork:



Figura B.3: iWork

Es la suite ofimática de aplicaciones creada por Apple, Inc., para el Sistema Operativo Mac OS X y el iOS.

Contiene un procesador de texto llamado “Pages”, un paquete de presentaciones llamado “Keynote” y una hoja de cálculo llamada “Numbers”. Aunque iWork fue en un principio un rumor de ser la actualización o reemplazo de la aplicación de Apple llamada “AppleWorks”, Apple actualmente la comercializa como “construyendo un sucesor de AppleWorks”, pero no extiende las funciones de bases de datos y herramientas de dibujo de AppleWorks. No pretende competir de forma directa con la versión para Mac de Microsoft Office que se considera más profesional; en cambio, ofrece herramientas con un punto de vista diferente, enfocado al usuario doméstico (presentación en vez de análisis) a un precio más accesible.

Está diseñado para integrarse con las aplicaciones de la suite iLife.

B.4 Pages:



Figura B.4: Pages

Pages es una aplicación de procesador de textos y layout desarrollada por Apple, Inc., como parte del set de productividad iWork (que también incluye “Keynote” y “Numbers”). Pages 1.0 fue anunciado a principios de 2005, y empezó a ser vendido en febrero de 2005. Pages 3 fue anunciado el 7 de agosto de 2007, solo para Mac OS X 10.4 “Tiger”, 10.5 “Leopard” y 10.6 “Snow Leopard”.

B.5 Keynote:**Figura B.5: Keynote**

Keynote es una aplicación de software de presentación desarrollada como parte del set de productividad iWork por Apple, Inc. Keynote 5 fue anunciado el 6 de enero de 2009, y es la versión más reciente. Se han añadido nuevos temas, transiciones y animaciones, así como la habilidad para controlar la presentación de diapositivas con un iPhone o iPod Touch por medio de la aplicación de Keynote remota, la cual está disponible en el iTunes Store.

B.6 Numbers:**Figura B.6: Numbers**

Numbers es una aplicación de hoja de cálculo desarrollada por Apple, Inc., como parte del set de productividad iWork (que también incluye Keynote y Pages). Numbers 1.0 fue anunciado el 7 de agosto de 2007, y funciona solo en Mac OS X v10.4 "Tiger" y Mac OS X v10.5 "Leopard". Numbers 2.0 fue anunciado el 6 de enero de 2009 con las nuevas características MathType y EndNote.

El principal competidor de Numbers es Microsoft Excel. Como una demostración de introducción, Steve Jobs lanzó una interfaz más fácil de usar, accesible para legos y que ofrece un mejor control sobre la apariencia, y la presentación de tablas de datos.

B.7 App Store:**Figura B.7: App Store (iPhone, iPod Touch y el iPad).**

App Store es un servicio para el iPhone, el iPod Touch, el iPad, Mac OS X Snow Leopard y Mac OS X Lion, creado por Apple, Inc., ya que permite a los usuarios buscar y descargar aplicaciones de iTunes Store o Mac App Store en el caso de Mac OS X, desarrolladas con el iPhone SDK y publicadas por Apple. Estas aplicaciones están disponibles para ser compradas o bien gratuitas, dependiendo de cada una. Las aplicaciones pueden ser descargadas directamente al iPhone o al iPod Touch por medio de una aplicación del mismo nombre, aunque el App Store también está disponible en el interior del programa informático iTunes.

iPhone SDK:

El lanzamiento del kit de desarrollo de software del iPhone fue anunciado en el evento iPhone 6 del año 2008. El SDK permite a los desarrolladores (que utilicen Mac OS X 10.5.4 o superior), crear aplicaciones usando el programa Xcode, que correrán nativamente en el iPhone y en el iPod Touch. Una versión beta fue lanzada después del evento y la versión final fue publicada en Julio de 2008, junto con el iPhone 3G. Este evento, acompañado con un gran programa de distribución para terceras personas, se convirtió más tarde en el Programa de Desarrolladores de iPhone que ofrece actualmente cuatro líneas de distribución para los desarrolladores:

- iOS Developer Program
- iOS Enterprise Program
- iOS University Program
- Mac Developer Program

Clasificación de Aplicaciones:

Apple clasifica las aplicaciones en base a su contenido, y para cada una determina para que grupo de edad es apropiada. Según el evento de lanzamiento del iPhone OS 3.0, el iPhone permitirá bloquear las aplicaciones desagradables en las opciones de éste. A continuación se mencionan las Categorías de Clasificación de Apple:

Clasificación	Descripción
4+	No contiene material desagradable.
9+	Puede contener situaciones leves o infrecuentes de violencia realista, fantástica o en dibujos animados frecuente, y contenido sugestivo, maduro o de terror que puede no ser apropiado para menores de 9 años.
12+	Puede contener lenguaje no apropiado leve o infrecuente, violencia realista, fantástica o en dibujos animados frecuente, y contenido maduro o sugestivo leve o no frecuente, y juegos de azar simulados que pueden no ser apropiados para menores de 12 años.

17+	Puede incluir contenido maduro, sugestivo o de terror intenso y frecuente; más contenido sexual o de desnudez, alcohol, tabaco y drogas que puede no ser apropiado para menores de 17 años. Los consumidores deben tener al menos 17 años para comprar aplicaciones con esta clasificación. Siempre que una aplicación con esta clasificación sea descargada, se mostrará un mensaje preguntando si el usuario tiene 17 años o más.
------------	---

Figura B.8: Tabla con la Clasificación de Edades para las Aplicaciones.

B.8 Mac App Store:



**Figura B.9: Mac App Store
(iMac, Macbook, MacBook Pro y MacBook Air).**

Mac App Store es una plataforma de distribución digital para el Sistema Mac OS X. Fue anunciada el 20 de octubre de 2010 en el evento “Back to Mac”, y está prevista para el 6 de enero de 2011. Apple empezó a recibir solicitudes de aplicaciones de desarrolladores registrados el 3 de noviembre de 2010.

Características:

Al igual que la App Store para el iPhone y el iPad, la Mac App Store se mantendrá siempre actualizada. Los desarrolladores tendrán la posibilidad de agregar nuevas funciones a sus aplicaciones para mejorar su funcionamiento o agregar nuevas funciones. Esta plataforma es un modelo adaptado que proviene de la App Store para dispositivos iOS y que permite a los usuarios:

- **Comprar, descargar e instalar** – La habilidad de comprar, descargar e instalar aplicaciones.
- **Volver a descargar** – Instalar aplicaciones previamente compradas otra vez en cada equipo que el usuario posea.
- **Actualizar** – La posibilidad de actualizar una o varias aplicaciones al tiempo.

B.9 iTunes Store:



Figura B.10: iTunes Store (iPhone, iPod Touch, iPad, iMac, MacBook, MacBook Pro y Macbook Air).

La iTunes Store es una tienda en línea de contenido digital de la empresa Apple, Inc., accesible únicamente a través del programa iTunes. La tienda "iTunes Music Store" abrió el 28 de abril de 2003 y sus productos poseían un sistema de gestión digital de derechos que impide la reproducción de los archivos comprados en computadoras que no tengan autorización (DRM). Actualmente el DRM ha sido eliminado de la tienda.

Características:

Precios:

Los precios por cada canción son variables, pero en general están alrededor de los 1.29 dólares en Estados Unidos, 99 centésimos de Euro en España y 13 pesos en México.

Disponibilidad:

iTunes Store:

iTunes Store se encuentra disponible en Alemania, Argentina, Australia, Austria, Bélgica, Bolivia, Brasil, Canadá, Chile, Chipre, Colombia, Costa Rica, Dinamarca, Ecuador, El Salvador, Estados Unidos, Finlandia, Francia, Guatemala, Grecia, Honduras, Irlanda, Italia, Japón, Luxemburgo, México, Nicaragua, Nueva Zelanda, Noruega, Países Bajos, Panamá, Paraguay, Perú, Portugal, República Dominicana, España, Suecia, Suiza, Reino Unido y Venezuela.

iTunes App Store:

Al 6 de agosto de 2009 iTunes App Store está disponible en estados Unidos, Argentina, Australia, Bélgica, Brasil, Canadá, Chile, China, Colombia, Costa Rica, Croacia, República Checa, Dinamarca, Alemania, Ecuador, España, Finlandia, Francia, Grecia, Guatemala, Hong Kong, Hungría, India, Indonesia, Irlanda, Israel, Italia, Korea, Kuwait, Líbano, Luxemburgo, Malasia, México, Nicaragua, Países Bajos, Nueva Zelanda, Noruega, Austria, Pakistán, Panamá, Perú, Filipinas, Polonia, Portugal, Qatar, Rumania, Rusia, Arabia Saudita, Suiza, Singapur, Eslovaquia, Eslovenia, Sudáfrica, Sri Lanka, Suecia, Taiwán, Tailandia, Turquía, Emiratos Árabes Unidos, Reino Unido, Venezuela, Vietnam y Japón.

iTunes Wi-Fi Music Store:

El lanzamiento del iPod Touch, trajo la introducción de la tienda iTunes Wi-Fi Music Store. Esta versión permite a los dueños de iPods, iPhones e iPads con capacidad inalámbrica Wi-Fi de adquirir música directamente en sus dispositivos portátiles de música.

Historia de iTunes Store:

Tras hacer su aparición el 28 de abril de 2003, iTunes Music Store fue la primer tienda de música en línea en ganar una amplia atención de la prensa. Las tiendas de Apple permiten al usuario adquirir canciones y transferirlas fácilmente a su dispositivo iPod a través del programa de computadora iTunes. El iPod es el único reproductor de música digital (además de algunos celulares Motorola y el iPhone), que está diseñado para trabajar con iTunes Store, aunque algunos otros reproductores de música digital trabajarán en el futuro con iTunes. iTunes Music Store se lanzó inicialmente con 200,000 archivos disponibles para descarga.

El 5 de septiembre de 2007, Apple introdujo el iPod touch que incluía el iTunes Wi-Fi Music Store. Una actualización del iPhone que incluía la tienda Wi-Fi Store fue lanzada el 28 de septiembre de 2007.

B.10 Bento 3 / Bento 4:



Figura B.11: Bento 3 / Bento 4

Bento es una base de datos que es fácil de utilizar, ya que gestiona los contactos, las tareas y demás información importante para el usuario. Como Bento es una base de datos, se pueden crear relaciones entre los datos y verlos de diferentes formas de las que se podrían ver en una hoja de cálculo.

Bento se ha diseñado para Mac OS X. La ventana de Bento tiene características que resultarán familiares, si ya se ha utilizado la Agenda, iCal, Keynote, iTunes o iPhoto.

Agrupación y Centralización:

Bento recopila toda la información importante del usuario en un único sitio para ayudar a organizarse. Se puede gestionar sus contactos, coordinar eventos, hacer seguimientos de proyectos, priorizar tareas, etc.

B.11 iDatabase:



Figura B.12: iDatabase para MacBook.



Figura B.13: iDatabase para el iPhone, iPod Touch y el iPad.

La Base de Datos personal básica y fácil de usar para todos. iDatabase permite catalogar toda la información que uno desea, para luego encontrarla fácilmente cuando se necesite. ¡Con este Gestor de Bases de Datos, se pueden administrar las Listas de Afiliación de diversos lugares a los que pertenezca uno, Gastos, Proyectos, hacer un seguimiento de las Cuentas de Internet y mucho más! Se pueden elegir de entre 22 Plantillas de Bases de Datos, que están listas para usar y que son fáciles de personalizar, para organizar cualquier tipo de información, e inclusive, se puede crear una Base de Datos totalmente nueva desde cero.

Plantillas Incluidas:

Se pueden elegir cualquiera de las 22 Plantillas de Bases de Datos, que están listas para usar que se pueden personalizar para satisfacer sus necesidades. A continuación se mencionan las Plantillas que están incluidas en este Programa: Cuentas, Libros, Colección de CD's, Clases, Computadoras, Contactos, Clientes, Colección de DVD's, Eventos, Diario de Ejercicios, Gastos, Inventario, Catálogo de Películas, Lista de Miembros de Afiliación de algún Lugar, Lista de Números de Teléfonos Celulares, Notas, Cuidado de Mascotas, Proyectos, Registros, Recetas, Para Hacer y Mantenimiento de Vehículos.

Campos Admitidos:

El tipo de Campo que se pueden crear para los Datos, son los siguientes: Texto pequeño, Texto Grande, Número, Fecha, Hora, Fecha y Hora, Imagen, Selector de Elementos, Contraseña, Casilla de Verificación, Enlace Web, Correo Electrónico y de Cálculo. También se puede utilizar un tipo de Campo Especial llamado "Separador de Campo", para dividir un Registro en distintas partes visuales.

Sincronización / Exportación / Importación y Copia de Seguridad:

iDatabase para Mac se puede sincronizar a través de Wi-Fi con iDatabase para el iPhone (se vende por separado). La aplicación soporta copia de seguridad y restauración, exportación e importación de base de datos con extensión .idatabase y .CSV. Los archivos .idatabse preservan el contenido completo y la estructura de la base de datos, incluidas las Imágenes, y también puede ser utilizado por la versión de iDatabase para el iPhone.

La Conexión a Internet no es Necesaria:

Las Características principales de la Aplicación no requiere ninguna Conexión a Internet.

ANEXO C: “HELP DESK”

C.1 Definición:

Es un sistema que se adapta a las necesidades de cada empresa, ya que permite resolver cualquier tipo de problema de forma ordenada, rápida y eficiente, además de que ayuda a organizar y controlar los activos de las Organizaciones logrando una mayor productividad corporativa con la consecuente reducción de costos de soporte. Reportes, estadísticas y gestión son las herramientas fundamentales para los responsables de solución (analistas).

El analista del Help Desk permite a los usuarios internos (empleados) y externos (clientes) ser los beneficiarios de la aplicación al conseguir una solución eficaz de los problemas a tiempo, llevando registros, reasignaciones y seguimientos del usuario reportado hasta la satisfactoria solución del problema.

C.2 Necesidad de un Sistema de Help Desk:

El sistema de Help Desk incrementa la productividad y aumenta la satisfacción de los usuarios internos y externos. Muchas empresas optan por delegar este servicio a terceros.

De nada sirve dotar a los empleados de una empresa con las más sofisticadas herramientas de productividad (computadoras, software y acceso a toda la información que esté disponible), si esto no viene acompañado de un detallado programa de capacitación y soporte que permita el real aprovechamiento de las Herramientas.

Tampoco sirve multiplicar el personal de soporte interno de una empresa para que los empleados tengan a quien acudir en cualquier momento. Estas prácticas son costos ocultos de la tecnología, ya que las empresas conocen bien esto y tratan de evitar a toda costa.

Por eso existen las herramientas de Help Desk, ya que son sistemas que permiten apoyar la labor productiva de los usuarios, garantizando la explotación eficiente de las aplicaciones, buscando a su vez la manera más eficiente y rentables de realizar esta Labor.

C.3 Beneficios dentro de la Organización:

La gestión de Help Desk, genera durante su práctica y ejecución continua, un número considerable de beneficios, los cuales se resumen a continuación:

- **Punto Único de Contacto.** Al tener un Punto Único de Contacto, el usuario obtiene asistencia inmediata por parte de las personas que tienen los conocimientos apropiados y la disposición para atenderlo.

- **Registro y Seguimiento de Problemas.** Con la gestión de Help Desk, se pretende crear estos mecanismos de forma automatizada el cual permita llevar un control preciso de todas las llamadas que se reciben, con la finalidad de generar en un determinado periodo de tiempo, mediciones que permitan conocer la razón de las llamadas y las soluciones propuestas.
- **Inventario de Hardware y Software.** Esto se hace con el propósito de conocer cuál es la capacidad instalada de inventario tecnológico (activos fijos), en dónde está siendo utilizado y por quién.
- **Responsabilidades y Funciones Definidas.** Uno de los Principios fundamentales de la gestión de Help Desk, es que deben constituirse equipos de trabajo con la responsabilidad de atender los problemas técnicos de los usuarios. Su función es buscar las soluciones que sean oportunas a los problemas presentados.
- **Productividad.** Este beneficio es uno de los más difíciles de demostrar a toda compañía que esté interesada en la función de Help Desk, aunque prácticamente es el más importante.

C.4 Justificación de un Modelo de Help Desk:

Con la variedad tan amplia de Modelos de Soporte Técnico o *Help Desk* (Mesa de Ayuda) que existen, es fácil perder de vista sus características elementales. De hecho, la mayoría del conocimiento y las habilidades que se requieren para manejar estos sistemas, son las mismas que se necesitan para manejar cualquier Organización. Las mismas líneas de acción aplicados al desarrollo, la administración y refinamiento de cualquier Departamento, frecuentemente también pueden ser usadas por el área de *Help Desk*.

Al igual que la mayoría de las Organizaciones, el sistema *Help Desk* funciona mejor cuando ha sido explícitamente diseñado (o por lo menos cuando su evolución ha sido controlada y planeada), que cuando se le permite un crecimiento sin control. Un sistema *Help Desk* explícitamente diseñado inicia con un modelo que muestra la forma de cómo el sistema proveerá los servicios y los recursos necesarios para hacerlo. En el caso de los sistemas ya existentes, el proceso debe incluir el análisis y la descripción del modelo que está siendo usado y después modificar este modelo para que se adapte a la nueva necesidad.

Para las Organizaciones que estén considerando crear un nuevo sistema de Soporte, el proceso debe comenzar con la pregunta de que si una Mesa de Ayuda es totalmente necesaria. Es posible que existan otras soluciones menos costosas o complicadas, pero que pueden trabajar igualmente bien. Por ejemplo, puede haber una mejor inversión a largo plazo, si se proporciona entrenamiento a los usuarios mientras estén dentro de la compañía, en lugar de crear una Mesa de Ayuda formal para resolver problemas.

En las Organizaciones que ya cuentan con un sistema *Help Desk*, el proceso de reingeniería o de reorganización frecuentemente es originado por situaciones externas a esta área, como son los cambios en los recursos disponibles, en la demanda (incluyendo modificaciones significativas en la tecnología) o en la percepción (real o de otra forma), de que el sistema no está cumpliendo con las necesidades de los usuarios, o no sea efectivo en costos.

C.5 Valoración de las Necesidades:

Para que el sistema satisfaga las necesidades de sus usuarios y de su compañía, éstas deben ser identificadas antes. El primer paso es reunir información acerca de las expectativas de aquellos que usarán el sistema y de las necesidades de soporte tecnológico dentro de la compañía.

Este proceso de recolección y evaluación de información se refiere a la valoración de las necesidades, el cual consiste en identificar correctamente qué servicios y recursos son apropiados para que el sistema pueda cumplir con su misión. En este contexto, se incluyen algunas consideraciones sobre su rendimiento, sobre que tan realista es el modelo de acuerdo con los recursos disponibles y las necesidades, y sobre los deseos de los usuarios y de la compañía.

La misión ayuda a establecer inicialmente el alcance de la valoración. Si la misión incluye dar soporte a toda la tecnología dentro de la compañía entonces es necesario reunir información sobre la tecnología, que está siendo usada en toda la Organización, incluyendo sistemas telefónicos, servicios de oficina y recursos de cómputo. Por otro lado, si la misión se enfoca a dar soporte sólo a las aplicaciones, y no a otras formas de tecnología (incluyendo *hardware*), entonces la evaluación debe enfocarse a la recolección de la información sobre el *software* que está siendo usado y las necesidades de soporte relacionadas con estas aplicaciones.

C.6 Consideraciones para un Nuevo Sistema de Help Desk:

Si un sistema formal *Help Desk* no existe dentro de la compañía, es importante reunir tanta información como sea posible sobre la forma en la que actualmente se realizan las actividades relacionadas con soporte. En las primeras etapas, tenga presente que uno de los elementos clave es determinar si los beneficios de un nuevo sistema de soporte serán redituables en costo. Para responder a esta pregunta, es muy importante identificar los costos del sistema informal. Algunas cuestiones que deben tomarse en cuenta son las siguientes:

- ¿Quién (o qué) está proporcionando ayuda a los usuarios?
- ¿Cuál es el costo del soporte informal en términos de la disminución de la productividad y otros recursos?

- Las personas que proveen de ayuda, ¿toman demasiado tiempo de su propio trabajo?
- ¿Cuál es el costo de no proporcionar soporte efectivo en términos de pérdida o desperdicio de esfuerzos y oportunidades?
- Además, intente saber qué es lo que sucede con las personas que necesitan Ayuda Técnica.
- ¿La tecnología nueva se está usando o se evita, y en su lugar se utilizan métodos anticuados? Si la nueva tecnología no se utiliza, ¿se debe a que la tecnología es inadecuada o a que las personas no saben cómo usarla apropiadamente?
- ¿Ha crecido la empresa reciente o ha cambiado significativamente la tecnología que se usa? ¿Existen cambios futuros significativos o necesidades de soporte anticipadas?

Si su Organización no cuenta en este momento con un sistema *Help Desk*, será difícil recopilar esta información. Hay que hablar con los usuarios para tener en cuenta sus percepciones, expectativas y preferencias en cuanto a los servicios de soporte se refiere.

También se tiene que recurrir a la Administración de la Empresa para obtener la información necesaria y, si existe, acudir al área de Sistemas.

Por último, es importante reunir la información sobre la tecnología que se está utilizando, y cómo se puede aplicar. Una empresa en la cual el personal utiliza una gran variedad de aplicaciones para programar, debe requerir un modelo muy distinto a una empresa en donde el procesador de palabras apenas ha comenzado a reemplazar la máquina de escribir.

C.7 Mecanismos de Reporte:

El mecanismo utilizado para reportar problemas al sistema *Help Desk* determina en gran parte cuál es el modelo que esté más apropiado. En general, los reportes de problemas llegan por medio de varias formas escritas (como Faxes, Cartas, Correo Electrónico u otros Mecanismos Electrónicos) o en formas verbales (Vía Telefónica o Contacto Personal). Cada método de reporte de problema tiene Ventajas y Desventajas, y cuentan con diferentes formatos con los cuales se pueden trabajar mejor en diferentes empresas, incluso para diferentes tipos de soporte o problemas.

Por supuesto, un sistema *Help Desk* no sólo recibe reportes en diferentes formatos, por lo que también es importante considerar la forma en que los diferentes mecanismos pueden interactuar. Parte del modelo debe incluir líneas de acción para cuando diferentes mecanismos de reportes sean usados o cuando se necesite priorizar entre diferentes reportes. Por ejemplo:

- ¿Recibe soporte primero alguien que acude directamente a *Help Desk* o alguien que habla?
- ¿Qué efecto tiene este tipo de política en el comportamiento del resto de los usuarios?
- ¿El reporte de problemas debe ser limitado a ciertas horas o a ciertos tipos de problemas?

Todas estas preguntas y muchas otras deben ser consideradas cuando se decide cómo deben ser reportados los problemas.

Algunas de las tecnologías más comunes para reportar problemas son discutidas aquí con sugerencias para una implementación efectiva.

Teléfono:

El contacto telefónico es uno de los medios más comunes y eficientes para reportar problemas. Las oportunidades para resolver el problema existen cuando el usuario que reporta el problema y el Ingeniero de soporte técnico interactúan directamente, pero ninguno necesita dejar su estación de trabajo.

Reporte en Línea:

El correo electrónico u otros métodos de Intercambio de información representan una manera conveniente de reportar problemas en los ambientes en donde la comunicación en línea ya es accesible. El uso de formas electrónicas y plantillas puede alentar al usuario a describir en forma consistente los síntomas del problema.

Fax:

El Fax sigue siendo una herramienta de comunicación para las empresas con operaciones en múltiples sitios, a pesar de la creciente aparición global de sistemas de correo electrónico. Su principal Ventaja es que los sistemas de Fax proporcionan una transferencia extremadamente rápida de documentos y comunicaciones escritas desde cualquier lugar del Mundo a otro. En situaciones en las cuales una forma estándar pueda ser proporcionada por adelantado a los usuarios, e incluya información apropiada, los Faxes serán una manera aceptable de reportar problemas. Como parte de un sistema de seguimiento de registros, los Faxes tienen un valor limitado debido a que normalmente requieren de copiado manual del historial del registro dentro del sistema de seguimiento, además de supervisión para los Faxes que lleguen.

Contacto Personal:

El reporte personal se usa básicamente cuando el sistema de soporte está situado dentro de la distancia de caminata de los usuarios. Si el reporte de problemas cara a cara es usado, el sistema debe asegurar que el problema será registrado y se llevará un seguimiento del mismo. El recurso de soporte apropiado debe entonces ser asignado para resolver el problema.

Una Desventaja del contacto personal es que puede causar conflicto, debido a que estos usuarios esperan normalmente una respuesta inmediata a sus problemas, y los recursos apropiados pueden no estar disponibles de inmediato. También resulta difícil priorizar los reportes.

ANEXO D: “WI-FI”

D.1 Conocimientos Básicos de Wi-Fi:

¿Qué es Wi-Fi?

Wi-Fi es una tecnología que permite que una gran variedad de equipos informáticos (computadoras, impresoras, discos duros, cámaras digitales, etc.) puedan interconectarse sin necesidad de utilizar cables. La aplicación principal que está teniendo Wi-Fi en la actualidad es la de permitir que varias computadoras que hay en los hogares, en las oficinas, etc., que puedan compartir el acceso a Internet (ya sea ADSL o cable). Sin embargo, esta tecnología permite crear una red entre los distintos equipos para compartir todos sus recursos.



Figura D.1: Logotipo Oficial de *Wireless Fidelity* (Wi-Fi).

Componentes de una Red Wi-Fi:

Una red Wi-Fi puede estar formada por dos computadoras o más, para que una de ellas pueda comunicarse de forma inalámbrica, necesita disponer de un dispositivo que se conoce como adaptador de red. Como se ha mencionado, un **adaptador de red** es un equipo (con transmisor, receptor y antena) que puede venir integrado en el equipo o instalado de forma independiente, y el cual le permite comunicarse de forma inalámbrica.

De forma general, a los equipos que forman parte de una red inalámbrica se les conoce como **terminales**.

Además de los adaptadores de red, las redes Wi-Fi pueden disponer también de unos equipos que reciben el nombre de **puntos de acceso** (AP o *Access Points*). Un **punto de acceso** es una estación base utilizada para gestionar las comunicaciones entre las distintas terminales de la red (entre los distintos adaptadores). Los puntos de acceso funcionan de forma autónoma, sin necesidad de estar conectados directamente a alguna computadora.

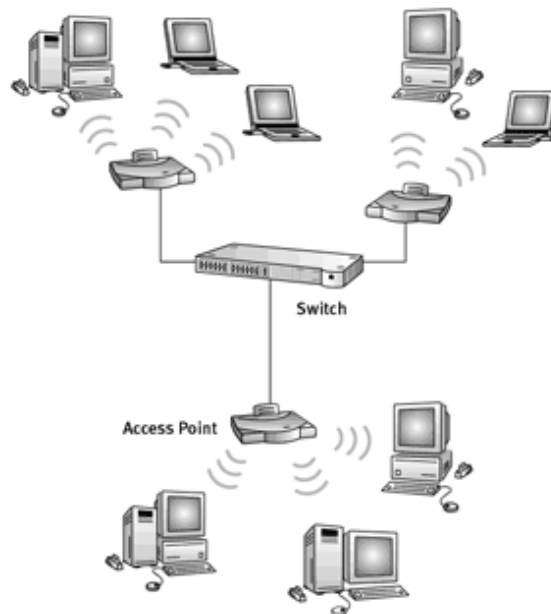


Figura D.2: Red Inalámbrica.

Tanto a las terminales como a los puntos de acceso se les conoce con el nombre de **estación**. Las estaciones se comunican entre sí, gracias a que utilizan la misma tecnología de comunicación.

Al conjunto de **protocolos** instalado en las estaciones se compone de dos grupos: uno que se ocupa de garantizar la comunicación inalámbrica entre las estaciones (que son los protocolos Wi-Fi), mientras que el otro se ocupa del intercambio de información entre las terminales, este último es de lo que se ocupan los protocolos TCP/IP. El hecho es que tanto Internet y las redes de cable también utilizan este protocolo para el intercambio de información, una terminal Wi-Fi no tiene dificultades para comunicarse con las terminales de estas otras redes.

Complementos de Wi-Fi (Bluetooth y 3G):

La Tecnología Bluetooth está pensada para conexiones de distancias cortas (menos de 10 mts.), Wi-Fi lo está para conexiones de área local (algunos cientos de metros) y las redes móviles para disponer de una cobertura global.

Bluetooth es una tecnología que, al contrario que Wi-Fi, no tiene por objetivo soportar redes de computadoras, sino más bien, para comunicar una computadora o cualquier otro dispositivo con sus accesorios complementarios que son los periféricos, por ejemplo: un teléfono móvil con su auricular, un PDA con su computadora, una computadora con su impresora, etc.

Habitualmente, la tecnología Bluetooth la incorporan las terminales de los teléfonos móviles para comunicarse con el "manos libres" del coche, intercambiar información con otras terminales o comunicarse con la computadora de la casa y/o de la oficina.

En cuanto al 3G o Tercera Generación de Telefonía Móvil, se trata de una tecnología que le ofrece a las terminales móviles la posibilidad de transmitir datos a alta velocidad. Mientras que con los móviles tradicionales no superan los 100 kbps reales (384 kbps Teóricos), con las terminales 3G se pretenden alcanzar varios Mbps. Independientemente de que la realidad se aparte bastante de la teoría, lo cierto es que ya se encuentran trabajando (inclusive se habla de la Red 4G) para conseguir una red de cobertura global que permita ofrecer un servicio de acceso a Internet a una velocidad aproximada de 100 Mbps.

Letras y Velocidades de Wi-Fi:

Wi-Fi es un estándar de comunicación inalámbrica. Si un fabricante desea construir un dispositivo Wi-Fi, sólo tendrá que seguir el estándar y listo. Los estándares son regulados por los Organismos de Normalización. Aunque existen muchos Organismos en el Mundo, el más relevante en el caso de Wi-Fi es el IEEE (*Institute of Electrical and Electronic Engineers*, “Instituto de Ingenieros Eléctricos y Electrónicos”). Si se accede a su Sitio Web, <http://standards.ieee.org/>, se podrán obtener los documentos técnicos de todas las Normas publicadas.

El estándar Wi-Fi está plasmado en la Norma IEEE 802.11b. Esta Norma describe los detalles técnicos para establecer comunicaciones de datos de forma inalámbrica a una velocidad máxima de 11 Mbps. Por tanto, el estándar original de Wi-Fi sólo permite transmitir datos a una velocidad máxima de 11 Mbps.

Por otro lado, la Tecnología de Comunicaciones Inalámbricas ha seguido desarrollándose, lo que ha permitido que posteriormente aparecieran diversas soluciones que permiten transmitir datos a velocidades superiores a 11 Mbps. Incluso se ha llegado a hablar de soluciones teóricas que alcanzan los 540Mbps. Muchas de estas nuevas soluciones llegan a tener su propia Norma IEEE. Las más interesantes son las siguientes:

- **IEEE 802.11b (1999).** Es la Norma original que permite velocidades de transmisión de hasta 11Mbps utilizando la banda de frecuencias de 2.4 GHz.
- **IEEE 802.11a (1999).** Esta Norma se diferencia de la Norma 802.11b en el hecho de que no utiliza la banda de los 2.4 GHz, sino la de 5 GHz. La gran ventaja es que se consiguen velocidades de 54 Mbps; llegando a alcanzar los 72 Mbps y hasta 108 Mbps con versiones propietarias (por ejemplo, la de Netgear).
- **IEEE 802.11g (2003).** Esta Norma surgió con la idea de aumentar la velocidad sin renunciar a las ventajas de la banda de los 2.4 GHz. Esta Norma permite transmitir datos a 54 Mbps. En cualquier caso, existen versiones propietarias que llegan a los 100 Mbps (por ejemplo, la de US Robotics).

- **IEEE 802.11n (2009)**. Se trata de un nuevo paso en el objetivo de conseguir velocidades cada vez mayores. Con esta Norma se habla de velocidades de 300 Mbps y alcances muchos mayores que con las Normas anteriores. Otra de las ventajas de esta Norma es que es compatible con todos los estándares antes mencionados (a, b y g). La característica externa más destacable de 802.11n es que incorpora varias antenas para poder utilizar varios canales simultáneamente.

La Anécdota del Origen de Wi-Fi:

La Tecnología Wi-Fi define una forma de intercambiar datos entre dos equipos utilizando ondas de radio. Esto es lo que se conoce como utilización del medio radioeléctrico. La tecnología básica en la que se fundamenta el funcionamiento de estos sistemas inalámbricos es el sistema conocido como "Espectro Expandido". Este sistema tiene la particularidad de ser muy resistente a las interferencias de otras fuentes de radio y a los efectos del eco, lo que le permite coexistir con otros sistemas de radiofrecuencia sin verse afectado fuertemente. Estas ventajas hacen que la tecnología de espectro expandido sea la más adecuada en las bandas de frecuencia en las que funciona Wi-Fi.

Curiosamente, la patente de las Técnicas de Espectro Expandido no la registraron los Militares, una Universidad o una Empresa de renombre, sino fueron la actriz austriaca Hedy Lamarr y el músico norteamericano George Antheil. Ambos desarrollaron una idea de un sistema secreto de comunicaciones en tiempos de la Segunda Guerra Mundial. Esta idea la patentaron en 1942. La idea se basaba en que, en vez de realizar una transmisión utilizando siempre la misma frecuencia, se usaran distintas frecuencias preestablecidas y hacer cambiar la frecuencia utilizada cada corto tiempo para evitar ser interceptados.

El problema de la idea de Lamarr y Antheil era que en su época no había la tecnología suficiente para ponerla en práctica. Aunque los militares norteamericanos le dieron uso en la década de los setentas, no ha sido hasta los años noventa, en la era de la tecnología de radio digital, cuando se le ha dado un Uso Comercial.

¿Por qué Instalar una Red Wi-Fi?

Se habla de red de computadoras (o de equipos informáticos) cuando varios de estos equipos están interconectados entre sí. Las grandes ventajas de poder contar con equipos interconectados son los siguientes:

- Permite compartir los servicios de comunicaciones. Fundamentalmente, el acceso a internet (mediante ADSL, módem cable, RDSI, etc.).
- Permite compartir periféricos: impresoras, scanners, discos duros en red, cámaras, etc.

- Permite compartir la información contenida en cada computadora.
- Permite compartir aplicaciones.

Se habla de distintos tipos de redes dependiendo de la extensión de la misma. Las redes locales son las que cubren el entorno de un edificio (casa, oficina, etc.), mientras que las redes metropolitanas, redes de área extensa o redes globales, cubren un entorno mayor: una ciudad, un país o un entorno multinacional.

Ventajas:

- Movilidad.
- Desplazamiento.
- Flexibilidad.
- Ahorro de Costos.
- Escalabilidad.

Desventajas:

- Interferencias.
- Cuanto mayores sean las interferencias producidas por otros equipos, menor será el rendimiento de la red.
- La velocidad de comunicación baja de forma automática hasta que se consigue una comunicación de calidad.

Soluciones de Telefonía:

La Telefonía Wi-Fi permite establecer y mantener conversaciones telefónicas utilizando sus facilidades de movilidad. Viene a ser la unión de las redes inalámbricas y de la telefonía IP.

La telefonía Wi-Fi puede tener su campo principal de aplicación en las empresas y sectores como la educación, salud, fabricación o almacenamiento, donde la movilidad de los trabajadores es un factor importante. La ventaja que ofrece la telefonía Wi-Fi frente a las comunicaciones inalámbricas de voz actuales es que permite integrar la facilidad de transmisión de voz con la de datos y video. Además, la existencia de lugares de acceso público Wi-Fi permite disponer de un servicio inalámbrico de bajo costo desde lugares públicos.



Figura D.3: Logotipo Oficial de la Zona de cobertura *Wi-Fi* que hay en Restaurantes, Centros Comerciales, Hospitales, Escuelas, etc.

Para poder hacer uso de esta tecnología, es necesario disponer de un aparato telefónico específico o de un dispositivo multipropósito (PDA o computadoras portátiles con software de IP). Las terminales basadas en PDA pueden resultar bastante interesantes, ya que permiten integrar las funciones de telefonía, video y datos en un solo equipo de tamaño reducido.

En este sentido, existen terminales telefónicos inalámbricos de tecnología Wi-Fi que permiten recibir y realizar llamadas telefónicas de voz, siempre y cuando se esté dentro del área de cobertura de una red Wi-Fi (dentro de la empresa, en la casa o conectado a alguna red Wi-Fi pública). Para evitar tener que llevar dos terminales distintos (uno Wi-Fi y otro de telefonía móvil), también hay terminales multimodos (GSM, Wi-Fi y 3G). Esto posibilita que, con una sola terminal, se puede utilizar Wi-Fi cuando se esté dentro del área de cobertura Wi-Fi y GSM o 3G cuando se esté fuera del área de cobertura Wi-Fi.

Otras Soluciones para la Empresa:

Desde el punto de vista de la empresa, las redes inalámbricas son aplicables en cualquier campo de la industria donde exista la necesidad de utilizar un equipo informático, tener movilidad y permanecer en contacto en tiempo real con recursos informáticos, ya sea dentro o fuera de la empresa.

Las redes inalámbricas son especialmente útiles cuando los empleados necesitan acceder a la información desde distintos sitios o mantener una cierta movilidad, por ejemplo: médicos, enfermeras, inspectores, agentes, vendedores, personal de mantenimiento, personal de almacén, atención al cliente, personal de exposición, etc.

Para ver las ventajas de las redes inalámbricas, sólo hay que pensar en la alternativa que existe actualmente como es utilizar formularios en papel, enseguida pasar la información recabada a la computadora, manejar información impresa que no está actualizada, tener que moverse para conseguir acceder a la Información de la empresa, etc. Esto por consiguiente, trae la duplicación del trabajo, aumentan los tiempos de respuesta, introducir errores de interpretación de la escritura manual, manejar información no actualizada, perder tiempo en desplazamientos innecesarios, etc.

Una computadora portátil o PDA con conexión inalámbrica puede mejorar bastante el rendimiento y la eficacia de muchos puestos de trabajo: facilita la movilidad, elimina el papeleo, disminuye los errores, reduce los costos de gestión, acerca de la empresa con el trabajador y aumenta la eficacia.

Son muchos los dispositivos móviles que se están incorporando a la tecnología Wi-Fi. En la mayoría de los casos, lo único que aporta esta tecnología es eliminar el cable, pero no faltan los ejemplos donde esa comunicación inalámbrica es el verdadero valor del equipo.

GLOSARIO DE TERMINOS TECNICOS

Acelerómetro:

Es un dispositivo que mide la aceleración y las fuerzas que son inducidas por la gravedad, ya que permite detectar el movimiento y el giro.

Aplicación:

Es un programa que está diseñado para asistir en la realización de una tarea específica, como puede ser un procesador de textos, contabilidad o gestión de inventario.

Aplicación Nativa:

Es un programa que está diseñado específicamente para un tipo en particular de microprocesador, es decir, un programa que es compatible a nivel binario con un procesador. Una aplicación nativa generalmente corre más rápido que una aplicación no nativa, que debe correrse con la ayuda de un programa de emulación.

Aplicación Web:

Son programas que se ejecutan en el entorno del navegador (por ejemplo, un applet de Java), o que esté codificado en algún lenguaje que soporte el navegador (como puede ser JavaScript combinado con HTML); para que así el navegador web pueda reproducir la aplicación.

Bit:

Es la unidad de información más pequeña, que es manejada por una computadora. Un bit expresa un 1 o un 0 en un número binario, o una condición lógica verdadera o falsa, y se representa físicamente por un elemento tal como un voltaje alto o bajo en un punto de circuito, o un punto pequeño de un disco magnetizado de una manera u otra.

Bluetooth:

Es una especificación tecnológica para redes inalámbricas, el cual permite la transmisión de voz y datos entre los distintos dispositivos móviles, mediante una radiofrecuencia segura (2.4 GHz). Esta tecnología permite las comunicaciones sin cables, ni conectores, así como la posibilidad de crear redes inalámbricas domésticas para sincronizar y compartir la información que se encuentra almacenada en diversos equipos.

Byte:

Es un dato que consta de 8 bits. Un byte puede representar un carácter único, como puede ser una letra, un dígito o una señal de puntuación.

C:

Es un lenguaje de programación desarrollado por Dennis Ritchie en los Laboratorios Bell en 1972. C es un lenguaje compilado que contiene un pequeño conjunto de funciones internas que son dependientes de la máquina. El resto de las funciones de C son dependientes de la plataforma y se contienen en librerías a las que se puede acceder desde programas desarrollados en C. Los programas en C están formados por una o más funciones definidas por el programador, por lo que se trata de un lenguaje estructurado.

C++:

Es una versión orientada a objetos del lenguaje de programación C, desarrollado por Bjarne Stroustrup en la década de los años 80, en los Laboratorios Bell, y fue adoptada por varios vendedores, incluyendo Apple, Inc., y Sun Microsystems, Inc.

Computadora:

Cualquier dispositivo capaz de procesar información para generar el resultado deseado. Sin importar su tamaño, las computadoras normalmente realizan su trabajo en tres pasos bien definidos:

1. Aceptar los datos de entrada.
2. Procesar la entrada según unas reglas predefinidas (programas).
3. Producir una salida.

Computadora Portátil (laptop):

Se denomina así a cualquier computadora diseñada para ser transportada con facilidad. Las computadoras portátiles se caracterizan por tamaño y peso.

CPU:

La unidad central de procesamiento es el dispositivo que interpreta y ejecuta instrucciones. Las unidades centrales de procesamiento de un solo chip, llamadas microprocesadores, hicieron posible las computadoras personales y las estaciones de trabajo.

Dato:

Es una unidad de información.

Dispositivo Móvil:

Son aparatos de tamaño pequeño, que tienen algunas capacidades de procesamiento, además de que cuentan con conexión, ya sea permanente o intermitente a una red, con memoria limitada, y están diseñados específicamente para una función, pero pueden llevar a cabo diversas funciones.

Hardware:

Son los componentes físicos de una computadora, incluyendo cualquier equipo periférico como las impresoras, módems y dispositivos señaladores, por ejemplo: el mouse.

IEEE:

Acrónimo de Institute of Electrical and Electronics Engineers. Organización de profesionales de la electrónica y la ingeniería basada en los Estados Unidos, pero que tiene miembros de muchos otros países. El IEEE se centra en temas eléctricos, electrónicos, sobre ingeniería de computadoras y científicos.

Informática:

Es el estudio de las computadoras, incluyendo su diseño, funcionamiento y uso en el procesamiento de información. La Informática combina tanto los aspectos teóricos como los prácticos de ingeniería, electrónica, teoría de la información, matemáticas, lógica y comportamiento humano. Los diversos aspectos de la Informática van desde la programación y arquitectura de la computadora a la inteligencia artificial y la robótica.

Interfaz:

Es un software que faculta a un programa para que pueda trabajar con el usuario (la interfaz de usuario, que puede consistir en una interfaz de línea de comandos, controlada por menús, o una interfaz gráfica), con otro programa, como el Sistema Operativo, o con el hardware de la computadora.

iOS:

Es un Sistema Operativo desarrollado por Apple, Inc., originalmente para el iPhone, pero le emplean además el iPod Touch, el iPad y el Apple TV. Posee una gran comunidad de desarrolladores de aplicaciones para este Sistema Operativo; además, de que es uno de los Sistemas Operativos más empleados en los "Smartphones", junto con Symbian y el BlackBerry.

Java:

Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems, Inc. Aunque es similar a C++, Java es más pequeño, portable y fácil de utilizar que C++, puesto que es más robusto y gestiona la memoria por sí mismo. Java también fue diseñado para ser seguro y neutro a las plataformas (lo que significa que se puede ejecutar sobre cualquier plataforma), por el hecho de que los programas en Java se compilan en bytecodes, que se asemejan al código máquina, y no son específicos de una plataforma, ejecutándose posteriormente en cualquier computadora que disponga de lo que se conoce como Máquina Virtual Java.

J2ME:

Es una colección de APIs en Java, que están orientadas a productos de consumo como los PDAs, teléfonos móviles o electrodomésticos. J2ME se ha convertido en una buena opción para crear juegos en teléfonos móviles, debido a que se puede emular en una PC durante la fase de desarrollo, y luego subirlos fácilmente al teléfono.

Kernel:

Conocido también como Núcleo, ya que es la parte esencial de un Sistema Operativo, el cual provee los servicios más básicos del Sistema. Se encarga de gestionar los recursos como es el acceso seguro al hardware de la computadora, además del multiplexado, el cual determina qué programa accederá a un determinado hardware, en caso de que dos o más quieran usarlo al mismo tiempo, y también ofrece una serie de abstracciones del hardware para que los programadores no tengan que acceder directamente al hardware, que este proceso puede llegar a ser complicado.

Macintosh:

Es la computadora personal desarrollada por Apple, Inc., fue puesta en el mercado en 1984, además de ser la primera en incorporar una interfaz gráfica. Su Sistema Operativo es el Mac OS.

Objective-C:

Es un lenguaje de programación orientado a objetos, que fue creado como un superconjunto de C, pero que implementa un modelo de objetos muy parecido al de Smalltalk. Originalmente fue creado por Brad Cox y la Corporación StepStone en 1980. En 1988 fue adoptado como lenguaje de programación de NEXTSTEP, y en 1992 fue liberado bajo la licencia GPL para el compilador GCC. Actualmente se usa como el lenguaje principal de programación en Mac OS X y GNUstep.

PDA:

Es una computadora palm de poco peso, que está diseñada para proporcionar funciones específicas como organización personal (calendario, notas, bases de datos, calculadora, etc.), así como también comunicaciones. Los Modelos más avanzados ofrecen también funciones Multimedia. Para el almacenamiento de los datos, las PDA cuentan con una memoria flash, en lugar de utilizar unidades de disco.

Procesamiento:

Es la manipulación de los datos de un Sistema de computadora. El procesamiento es el paso vital entre recibir datos (entrada), y producir resultados (salida), la tarea para la que han sido diseñadas las computadoras.

Programación:

Es la creación de programas de computación, ya que requiere el conocimiento de uno o más lenguajes de programación, tales como Basic, C, Pascal, Lenguaje Ensamblador. Conocer únicamente un lenguaje de programación no implica la creación de buenos programas. Hay otros factores que se deben de considerar como es el conocimiento de la teoría de algoritmos, el diseño de la interfaz de usuario y el conocimiento de las características de los dispositivos de hardware. Las computadoras son máquinas lógicas y la programación también tiene que realizarse desde una aproximación lógica para poder diseñar, escribir (codificar), probar y depurar un programa.

Red:

Es un grupo de computadoras y dispositivos asociados que son conectados para tener facilidades de comunicaciones, ya que puede implicar conexiones permanentes, como cables, o las conexiones temporales realizadas a través del teléfono u otros enlaces de comunicación; además, una red puede ser pequeña como por ejemplo una red de área local, la cual consiste en pocas computadoras, impresoras y otros dispositivos, o pueden llegar a consistir en varias computadoras, tanto pequeñas como grandes, que se encuentran distribuidas en un área geográfica extensa.

SDK:

Abreviatura de Kit de Desarrollo de Software. Es un conjunto de rutinas (normalmente de una o más bibliotecas) que están diseñadas para permitir a los desarrolladores escribir programas más fácilmente, ya sea en una determinada computadora, sistema operativo o interfaz de usuario.

Servidor:

Es una computadora o programa que responde a las órdenes enviadas desde un cliente. Por ejemplo: un servidor de archivos puede contener archivos de datos o de programa; cuando un cliente envía una solicitud para importar un archivo, el servidor transfiere al cliente una copia del archivo solicitado.

Sistema:

Es una colección de elementos que trabajan conjuntamente para llevar a cabo una tarea. Por ejemplo: un Sistema Hardware consta de un microprocesador, sus chips complementarios y demás circuitos, dispositivos de entrada y salida, así como los periféricos; un Sistema Operativo que consta de un grupo de programas y archivos de datos; o un Sistema de Gestión de Bases de Datos, el cual es utilizado para procesar tipos específicos de información.

Sistema Distribuido:

En una red no centralizada que consta de varios equipos que se pueden comunicar entre sí, y que aparecen ante el usuario como parte de un sencillo, grande y accesible almacén de hardware, software y datos que se pueden compartir.

Sistema Operativo:

Es un software que controla la ubicación y el uso de los recursos de hardware como son la memoria, el tiempo de la unidad central de procesamiento (CPU), el espacio de disco, así como también los dispositivos periféricos. El Sistema Operativo es la base sobre la que se construyen las aplicaciones. Entre los Sistemas Operativos más populares se encuentran Windows (cualquiera de sus versiones), Mac OS y UNIX.

Smartphone:

Es un teléfono móvil que incorpora diversas características de una computadora personal, suelen permitir al usuario instalar nuevas aplicaciones, aumentando así sus funcionalidades, esto es posible, debido a que ejecutan un Sistema Operativo potente de fondo, además de que tienen un mini teclado, una pantalla táctil, cuentan con acceso a Internet, revisar correos electrónicos, cámara integrada, navegador web, procesador de textos, etc.

Software:

Son programas informáticos que contiene a su vez, las instrucciones que hacen que funcione el hardware. Dos de los principales tipos de software del sistema (sistemas operativos), que controlan los trabajos ejecutados en una computadora, y las aplicaciones, como pueden ser los procesadores de texto, las hojas de cálculo y las bases de datos, ya que realizan las tareas que justifican la utilización de las computadoras.

Soporte Técnico:

Es una asistencia, como puede ser el asesoramiento técnico que es proporcionado directamente hacia los clientes.

Tableta:

Es un tipo de computadora portátil, de mayor tamaño que un teléfono inteligente o un PDA, integrado con una pantalla táctil con la que se interactúa primariamente con los dedos o con una *pluma stylus*, sin necesidad de tener un teclado físico, ni ratón. Estos últimos se ven reemplazados por un teclado virtual y, en determinados modelos, por un mini-trackball intergrado en uno de los bordes de la pantalla.

UNIX:

Es un sistema operativo multiusuario multitarea, que fue desarrollado por Ken Thompson y Dennis Ritchie en AT&T Bell Laboratories en 1969, para su uso en minicomputadoras. Se considera que UNIX es un sistema operativo potente que, como está escrito en el lenguaje C, es más portátil, es decir, menos específico de la máquina que otros sistemas operativos.

Web:

Es un conjunto de documentos, que están entrelazados en un sistema de hipertexto. El usuario entra a la web, a través de una página de inicio.

Wi-Fi:

Es un conjunto de estándares para redes inalámbricas, el cual está basado en las especificaciones IEEE 802.11 (especialmente la 802.11b y la 802.11g), creado para redes locales inalámbricas, además de que se utiliza también para tener acceso a Internet.

Xcode:

Es el entorno de desarrollo integrado de Apple, Inc., es el motor que le brinda poder al ambiente integrado de desarrollo de Apple para Mac OS X y para el iPhones OS. Es completamente funcional para el desarrollo de aplicaciones Mac e incluye un editor de código de clase mundial, un depurador gráfico, e integrado en Objective-C, C y C++ compiladores, ya que posee un profundo conocimiento de los marcos que el poder del Cocoa Mac OS X, e incluso es capaz de identificar los errores mediante el análisis del código que se escribe (sin ejecutar la aplicación). La experiencia de desarrollo intuitiva hace que sea fácil de obtener su primera aplicación en funcionamiento, sin embargo, Xcode es tan poderoso que el propio Mac OS X el cual se encuentra construido, ya que utiliza su conjunto de herramientas.

BIBLIOGRAFIA

Libros:

- Asteasuain, Fernando. **UML**. Ed. Desarrolladores USERS. Primera Edición. Argentina. 2009. 320 pp.
- Carballar, José Antonio. **Wi-Fi: Lo que se necesita conocer**. Ed. Alfaomega. Primera Edición. México. 2010. 212 pp.
- Microsoft. **Diccionario de Informática e Internet**. Ed. Mc Graw-Hill. Primera Edición. España. 2003. 670 pp.

Libros Electrónicos (PDF):

- Allen, Grant, et.al. **The Definitive Guide to SQLite**. Ed. Apress. Second Edition. United States of America. 2010. 350 pp.
- Alessi, Patrick. **Professional iPhone® and iPad™ Database Application Programming**. Ed. Wiley Publishing, Inc. First Edition, United States of America, 2011, 378 pp.
- Ray, John. **Sams Teach Yourself: iPad™ Application Development in 24 Hours**. Ed. SAMS. First Edition. United States of America. 2010. 684 pp.
- Lee, Wei-Meng. **Beginning iPad™ Application Development**. Ed. Wiley Publishing, In. First Edition. United States of America. 2010. 566 pp.
- Pilone, Dan, et.al. **Head First iPhone® Development**. Ed. O'Reilly. First Edition. United States of America. 2010. 550 pp.
- Allan, Alasdair. **Learning iPhone® Programming**. Ed. O'Reilly. First Edition. United States of America. 2010. 358 pp.
- Lewis, Dr. Rory. **Aplicaciones iPhone® e iPad™ para Principiantes**. Ed. Apress. Primera Edición. Estados Unidos de América. 2010. 318 pp.
- Dalrymple, Mark, et.al. **Learn Objective-C on the Mac**. Ed. Apress. First Edition. United States of America. 2009. 362 pp.
- Ali, Maher. **Advanced iOS 4 Programming: Developing Mobile Applications for Apple iPhone®, iPad™ and iPod® Touch**. Ed. Wiley. First Edition. United States of America. 2011. 698 pp.

Tutoriales:

- Diseño de interfaces de Usuario con XCODE.
- Programación Orientada a Objetos.
- Xcode 4 Transition Guide (Apple Official Guide).
- Historia de Apple, Inc.
- Programar en Mac (AppleNeXt).
- Introducción al Entorno de Programación de Mac OS X.
- Guía iPhone® (Maestros del web).
- Uso Básico de Xcode & Interface Builder en Mac OS X usando la librería Cocoa de Objective-C.
- El Lenguaje Objective-C para Programadores C++ y Java.
- Objective-C and Cocoa User Guide and Reference Manual.
- The Objective-C Programming Language (Apple Official Guide).
- Cocoa Application Tutorial (Apple Official Guide).
- Programming in Objective-C 2.0.
- A Tour of Xcode (Apple Official Guide).
- Become an Xcoder.

Video-Tutoriales:

- Tutorial for Xcode (30 Videos).
- Tutorial for Cocoa (28 Videos).
- Tutorial for Objective-C (20 Videos).

Internet:

- <http://www.iphonedevcentral.com/>
- <http://www.maniacdev.com/ipad-development-tutorial/>
- <http://www.techotopia.com/>
- <http://www.roseindia.net/tutorial/iphone/examples/index.html>
- <http://www.bogotobogo.com/index.html>
- <http://www.theappcodeblog.com/>
- <http://www.apple.com/mx/>
- <http://www.muycomputerpro.com/2011/08/17/razon-i-apple/>
- <http://revista.seguridad.unam.mx/numero-07/dispositivos-moviles>
- <http://www.colombiadigital.net/entorno-tic/especial-del-mes/dispositivos-moviles.html>
- http://www.iabmexico.com/usosyhabitos_mobile
- <http://www.eluniversal.com.mx/articulos/63109.html>
- <http://www.air-watch.com/es/>
- http://www.unesco.org/new/es/education/resources/online-materials/single-view/news/learning_with_mobile_technologies/