

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN - MANAGUA
RECINTO UNIVERSITARIO RUBÉN DARÍO
FACULTAD DE CIENCIAS E INGENIERÍAS
DEPARTAMENTO DE COMPUTACIÓN



**Seminario de graduación para optar al título de Licenciatura en Ciencias de la
Computación**

Tema:

Programación de Aplicaciones para Móviles.

Tema delimitado:

Aplicación móvil bajo la plataforma Android para la suscripción y notificaciones de la Asociación de Emprendedores de la comunidad de renovación familiar “HOSANNA”, año 2014.

Elaborado por:

➤ Br. Rosa Marina Sequeira

Tutor:

Lic. Roberto Ezequiel Morales Suárez

Dedicatoria

A:

DIOS:

Gracias doy a mi Dios quien me ha dado la oportunidad de culminar esta etapa de mi formación profesional, la fortaleza para no desmayar y quien siempre me sustenta y me da la sabiduría.

A MI FAMILIA:

Por todo el apoyo y la comprensión durante mis estudios y culminación de mi tesis de seminario.

A MAESTROS(AS):

Que de una u otra manera colabore con mi preparación profesional hasta culminar con la preparación y defensa de esta tesis.

(Br. Rosa Marina Sequiera Palma.)

Agradecimiento

A Dios por darme la oportunidad de finalizar esta tesis de seminario, por ser mi luz, mi fortaleza y guía en todo momento.

A mis padres y esposo que en todo momento me han apoyado, por su amor, sus valores, motivaciones y sabiduría.

A Mi tutor Lic. Roberto Ezequiel Morales Suarez por su calidad humana profesional y colaboración.

(Br. Rosa Marina Sequeira Palma)

INDICE

I. RESUMEN	6
II. INTRODUCCION	7
III. JUSTIFICACION	8
IV. ANTECEDENTES	9
V. OBJETIVOS	10
VI. MARCO TEORICO	11
6.1 Conceptos básicos de programación	11
6.1.1 Programación	11
6.1.2 Lenguaje de Programación.....	11
6.1.3 Programación Orientada a Objetos	11
6.1.4 Algoritmo	11
6.2 Servicios Web	12
6.2.1. Tecnología en Servicios Web	12
6.3 Sistema Operativo.	13
6.3.1 Funciones básicas del S.O.....	13
6.4 Introducción a las Tecnologías móviles	14
6.4.1 Dispositivos móviles	14
6.4.2 Tipos de dispositivos móviles	14
6.4.3 Sistemas Operativos para dispositivos móviles.....	16
6.4.3.1 Características básicas de cada S.O:	16
6.4.4 Aplicaciones Móviles	18
6.5 Android	19
6.5.1 ¿Que es Android?	19
6.5.2 Principales Características	19
6.5.3 Arquitectura de Android	20
6.5.4 Herramientas de desarrollo	24
6.5.5 Estructura de un Proyecto Android	25
6.5.6 Componentes de una aplicación Android.....	28
6.5.7 Versiones de Android y Niveles de API	30
6.5.7.1 Características de las principales versiones.	30
6.6 Java ME	32
6.6.1 Java.....	32
6.6.2 Ediciones de Java 2.....	32
6.6.3 Arquitectura Java ME.....	33

6.6.4 Librerías	34
6.6.5 Comunicaciones en MIDlets	35
6.7 Análisis y Diseño de Sistemas	36
6.7.1. Modelos de Desarrollo de Sistemas	36
6.7.1.1 ¿Qué es un modelo de desarrollo?	36
6.7.1.2 Modelos Desarrollo Evolutivos.....	37
6.7.1.3. Prototipo.	42
6.7.2. Metodología de Investigación.....	44
6.7.3. Métodos de recolección de datos	45
6.7.4. Base de Datos.....	46
6.7.4.1. Base de Datos Relacional.....	46
6.7.4.2. Normalización de Base de Datos.....	48
6.7.5. Diccionario de Datos.....	50
6.7.6. SQL (Structured Query Language)	51
6.7.7. UML (Lenguaje Unificado de Modelado).....	52
6.7.7.1. Diagramas de Clases	53
6.7.7.2. Diagramas de Caso de Uso.....	54
6.7.7.3. Diagramas de secuencias.....	55
6.7.7.4. Diagramas de colaboración	55
6.7.8 Estudio de Factibilidad	56
6.8 Asociación de Emprendedores Hosanna.....	57
VII. DISEÑO METODOLOGICO	59
7.1 Tipo de Investigación.....	59
7.2 Población y Muestra.	59
7.3 Técnicas e instrumentos de recolección de datos.	59
7.3.1 Técnicas	59
7.3.2 Instrumentos	60
7.4 Modelo de desarrollo.	60
VIII. ANALISIS DE RESULTADO	62
8.1 Encuesta.....	62
8.2 IDE (Entorno de Desarrollo Integrado)	64
8.3 Distribución de las versiones del OS Android.....	64
8.4 Servicio Web.	66
8.5 Herramientas Utilizadas.	67
8.6 Descripción de la aplicación.	68
8.7 Estudio de Factibilidad.....	69
8.7.1 Factibilidad Técnica.....	69
8.7.2 Factibilidad Económica	71
8.7.3 Factibilidad Operacional.....	72

8.8 Diagramas UML.	73
8.8.1. Diagramas de Caso de USO	73
8.8.2. Diagramas de Secuencia	76
8.8.3. Diagrama de Clases.....	80
8.8.4 Diagrama de Colaboración	81
8.8.5 Diagrama E/R	83
8.9 Diccionario de Datos	84
8.10 Pantallas de la aplicación.....	87
8.11 Pruebas realizadas.....	89
IX. CONCLUSIONES	90
X. RECOMENDACIONES	91
XI. BIBLIOGRAFIA	92
XII ANEXOS	94

I. RESUMEN

Los dispositivos móviles constituyen cada vez más una realidad que ofrece al usuario, en un mismo y reducido aparato, funciones de comunicación y procesamiento de datos que van mucho más allá de las simples llamadas telefónicas o la ejecución de aplicaciones básicas. El gigante de Internet Google ha presentado un nuevo sistema operativo para este tipo de dispositivos, Android, que busca ser una firme alternativa a otros sistemas ya ampliamente extendidos como Symbian o Windows Mobile.

El presente proyecto sobre el desarrollo de una aplicación móvil bajo la plataforma Android para la suscripción y notificaciones de la Asociación de Emprendedores de la comunidad de renovación familiar “HOSANNA”, 2014, fue realizado con el propósito de realizar de una manera más eficaz y novedosa la suscripción a esta asociación, mejorando el canal de comunicación manteniendo informados en todo momento a sus miembros sobre las diferentes actividades o programas que esta escuela de negocios realiza.

El tipo de investigación efectuada fue Aplicativa, ya que se han implementado un conjunto de conocimientos teóricos y prácticos adquiridos la etapa de formación como profesionales en una nueva plataforma de desarrollo y sistema operativo móvil, Android, basado en Linux y Java

Durante el proceso de selección y obtención de información se utilizaron como herramientas la encuesta y la entrevista con el fin de valorar la aceptación y viabilidad del proyecto.

II. INTRODUCCION

Poco tiempo ha transcurrido desde la salida al mercado de los teléfonos inteligentes (también llamados Smartphone); dispositivos que no solo permitían hacer llamadas o enviar mensajes, sino que podían realizar tareas más complejas como administrar correos, organizar agendas, navegar por Internet, entre otras.

En un principio, estos dispositivos fueron considerados como un artículo de lujo al que solo grandes empresarios o directivos podían tener acceso; sin embargo; con el paso del tiempo este paradigma se rompió y los dispositivos móviles se han convertido en una de las principales herramientas de entretenimiento, comunicación y productividad para la población en general.

Es gracias al aumento en el uso de dispositivos móviles que los expertos y entusiastas de la tecnología vieron el potencial de dicha tecnología en el desarrollo de soluciones orientadas a la movilidad; naciendo de esta forma el nuevo campo del desarrollo de aplicaciones para dispositivos móviles (mejor llamadas Apps).

Muchas Comunidades Religiosas alrededor del mundo están aprovechando el auge de las teléfonos inteligentes, para incorporar estas nuevas tecnologías a la iglesia y a la vida de sus fieles y de esta manera mantenerles informados de todas las actividades, eventos y servicios, permitiéndoles involucrarse en todo momento y en cualquier lugar.

Asociación de Emprendedores de la comunidad de renovación familiar HOSANNA es una asociación Cristo céntrica conformada por emprendedores, empresarios y profesionales enfocados en la restauración y crecimiento de las economías de la empresas y familias, mediante la implantación de los principios establecidos en la palabra de Dios.

Esta Asociación ha decidido aprovechar los avances de las nuevas tecnologías de telefonía móvil para mantener una comunicación más directa y rápida con sus miembros desarrollando una Aplicación móvil que permita notificar a sus miembros sobre sus diferente eventos, así como garantizarles un método de suscripción más eficiente.

III. JUSTIFICACION

El presente proyecto realizara una aplicación móvil para la asociación de emprendedores de la Comunidad Hosanna, bajo la plataforma Android, con el propósito de realizar de una manera más eficaz y novedosa la suscripción a esta asociación y mantener informados en todo momento a sus miembros sobre las diferentes actividades o programas que esta escuela de negocios realiza.

En la actualidad la Asociación de Emprendedores Hosanna realiza sus notificaciones través de mensajes masivos y/o correo electrónico a cada miembro de la asociación, da a conocer sus programas a través de una página web estática y boletines semanales de la Comunidad Renovación Familiar Hosanna lo cual es una limitante ya que esta escuela de negocios no quiere restringirse solo a los miembros y visitantes de la iglesia.

Con el desarrollo e implementación de la aplicación móvil bajo la plataforma Android para la suscripción y notificaciones de la Asociación de Emprendedores de la Comunidad de Renovación familiar Hosanna se mejorara el canal de comunicación con cada uno de sus miembros permitiendo que siempre estén actualizados sobre la información de los programas y conferencias que esta escuela de negocios brinda a de más de que la realización de suscripciones que permitirá a la asociación tener el registro actualizado de las personas que a través de esta aplicación están siguiendo las actividades que se realizan y tendrá mayor presencia en las terminales de cada usuario permitiendo a su vez un mayor posicionamiento de marca y alcance de miembros potenciales que visiten la plataforma de distribución.

IV. ANTECEDENTES

Las primeras aplicaciones móviles que se desarrollaron datan de finales de los 90s; con la aparición de los primeros Smartphone, estas eran lo que conocemos como la agenda, contactos, editores de ring tones, etc. Dichas aplicaciones cumplían con funciones muy elementales y su diseño era bastante simple y poco atractivo. (aplicaciones móviles la evolución, 2013)

La evolución de las App se dio rápidamente gracias a las innovaciones en tecnología WAP y la transmisión de datos (EDGE) esto vino acompañado de un desarrollo muy fuerte de los celulares.

Finalmente la evolución de dichas aplicaciones nace con el lanzamiento del iPhone de Apple y el desarrollo del sistema operativo para móviles Android. Junto a estos desarrollos llegan muchas más propuestas de Smartphone, y de esta forma empieza el boom de las App, juegos, noticias, diseño, arte, educación, fotografía, medicina todo inmerso en lo que antes eran un simple equipo de comunicación celular, la incorporación de internet en los celulares y la creación de las Tablets revolucionó el mundo de las aplicaciones móviles. Ahora es normal de ver a personas de cualquier edad y estatus utilizando aplicaciones, y es por que poco a poco estos aparatos se están volviendo imprescindibles, ya que es una realidad que hoy en día las personas, entran a internet desde su dispositivo móvil que desde sus computadoras. (Aplicaciones Móviles, La Evolución, 2013)

Aprovechando este hecho es que los negocios alrededor del mundo en la actualidad están incorporando a sus negocios aplicaciones móviles que les permiten posicionamiento de marca, promocionar productos, implicar a los clientes con un nuevo punto de contacto, llegar a ellos donde se encuentren entre otros.

En Nicaragua los únicos negocios que están aprovechando esta tecnología son los de comunicación y transporte colectivo que actualmente está limitado para Managua.

V. OBJETIVOS

5.1 Objetivo General

Desarrollar una Aplicación móvil bajo la plataforma Android para la suscripción y notificaciones de la Asociación de Emprendedores de la comunidad de renovación familiar “HOSANNA”, año 2014.

5.2 Objetivos Específicos

- Analizar las diferentes Tecnologías y herramientas utilizadas en el diseño y desarrollo de aplicaciones móviles.
- Evaluar las tecnologías y herramientas utilizadas en el diseño y desarrollo de aplicaciones móviles para su implementación en la suscripción y notificaciones de Asociación de Emprendedores.
- Diseñar la aplicación móvil utilizando la plataforma Android.
- Programar la aplicación para la plataforma Android utilizando el lenguaje de programación java y los aspectos básicos del desarrollo orientada a objetos.

VI. MARCO TEORICO

6.1 Conceptos básicos de programación.

6.1.1 Programación

Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado. (Blogspot, 2010)

En un plano más técnico, la programación se realiza mediante el uso de algoritmos, que son secuencias finitas, ordenadas y no ambiguas de instrucciones que deben seguirse para resolver un problema. (Definicion.de, 2010)

6.1.2 Lenguaje de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. (Mcornielly, 2009)

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. (Mcornielly, 2009)

6.1.3 Programación Orientada a Objetos

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. (Mcornielly, 2009)

6.1.4 Algoritmo

Un Algoritmo, se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un

problema. Por lo tanto podemos decir que es un conjunto ordenado y finito de pasos que nos permite solucionar un problema. (Frida, 2009)

Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje de diferente programación. (Frida, 2009)

6.2 Servicios Web

El término Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para etiquetar los datos, SOAP para transferir los datos, WSDL se emplea para describir los servicios disponibles y UDDI para listar los servicios disponibles.

Un servicio Web es un mecanismo que permite llamadas remotas a métodos a través de HTTP mediante el uso de protocolos y lenguajes basados en XML, que ofrecen una sencilla forma de extender la funcionalidad de aplicaciones empresariales.

6.2.1. Tecnología en Servicios Web

Los Servicios Web están contruidos con varias tecnologías que trabajan conjuntamente con los estándares que están emergiendo para asegurar la seguridad y operatividad, de modo de hacer realidad que el uso combinado de varios Web Services, independiente de la o las empresas que los proveen, esté garantizado. A continuación se describen brevemente los estándares que están ocupando los Web Services. (Saffirio, 2006)

XML (Extensible Markup Language)

El XML (**Lenguaje de Marcado Extensible**) es una especificación desarrollada por W3C. Permite a los desarrolladores crear sus propias etiquetas personalizadas, que les permiten habilitar definiciones, transmisiones, validaciones, e interpretación de los datos entre diversas aplicaciones y organizaciones. (Saffirio, 2006)

SOAP (Simple Object Acces Protocol)

Es un protocolo de comunicación, basado en XML que se usa para codificar información de los requerimientos de los web services y para responder los mensajes antes de enviarlos por la red. Los mensajes SOAP son independientes de los sistemas operativos y pueden ser transportados por los protocolos que funcionan en el internet como pueden ser SMTP, MIME y HTTP. (Saffirio, 2006)

WSDL

Abreviación de Web Services Description Language, es un lenguaje especificado en XML que se ocupa para definir los Web Services como colecciones de punto de comunicación capaces de intercambiar mensajes. El WSDL es parte integral de UDDI y parte del registro global de XML, en otras palabras es un estándar de uso público (no se requiere pagar licencias ni royalties para usarlo). (Saffirio, 2006)

UDDI

Abreviación de Universal Description Discovery and Integration. Es un directorio distribuido que opera en la web que permite a las empresas publicar sus Web Services, para que otras empresas conozcan y utilicen los web servicios que publican, operan de manera análoga a las páginas amarillas. (Saffirio, 2006)

6.3 Sistema Operativo.

Es el conjunto de programas informáticos que permite la administración eficaz de los recursos de una computadora comienza a trabajar apenas se enciende el equipo, ya que gestionan el hardware desde los niveles más básicos y permiten además la interacción con el usuario. (Definicion.de, 2010)

Cabe destacar que los sistemas operativos no funcionan sólo en las computadoras. Por el contrario, este tipo de sistemas se encuentran en la mayoría de los dispositivos electrónicos que utilizan microprocesadores: el software de sistema posibilita que el aparato cumpla con sus funciones (por ejemplo, un teléfono móvil o un reproductor de DVD). . (Definicion.de, 2010)

6.3.1 Funciones básicas del S.O.

El sistema operativo cumple con cinco funciones básicas:

- El suministro de interfaz al usuario.
- La administración de recursos.
- La administración de archivos.
- La administración de tareas.
- El servicio de soporte y utilidades.

(Definicion.de, 2010)

6. 4 Introducción a las Tecnologías móviles.

6.4.1 Dispositivos móviles

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablets PCs. En este trabajo nos centramos fundamentalmente en los teléfonos móviles y en los PDAs por ser los tipos de dispositivos más utilizados y conocidos en la actualidad, los que ofrecen mayor variedad de aplicaciones multimedia y los que más posibilidades de evolución presentan en este sentido. (Baz Alonso, Ferreira Artime, Alvarez Rodrigues, & Garcia Baniello, 2009)

6.4.2 Tipos de dispositivos móviles

6.4.2 .1 PDAs (Personal Digital Assistant)

Un PDA, es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Hoy día estos dispositivos, pueden realizar muchas de las funciones de una computadora de escritorio pero con la ventaja de ser portátil. Inicialmente los PDAs incluían aplicaciones estrictamente relacionadas con su función como agenda electrónica, es decir, se reducían a calendario, lista de contactos, bloc de notas y recordatorios. Con el paso de tiempo han ido evolucionando hasta los dispositivos actuales que ofertan un rango mucho más extendido de aplicaciones, como juegos, acceso al correo electrónico o la posibilidad de ver películas, crear documentos, navegar por Internet o reproducir archivos de audio. (Baz Alonso, Ferreira Artime, Alvarez Rodrigues, & Garcia Baniello, 2009)

Las características del PDA moderno son pantalla sensible al tacto, conexión a una computadora para sincronización, ranura para tarjeta de memoria, y al menos Infrarrojo, Bluetooth o WiFi. La irrupción de Microsoft Windows CE (2000) y Windows Mobile (2003) en el sector los dotó de mayores capacidades multimedia y conectividad. Las PDAs de hoy en día traen multitud de comunicaciones inalámbricas (Bluetooth, WiFi, IrDA, GPS,) que los hace tremendamente atractivos hasta para cosas tan inverosímiles como su uso para demótico o como navegadores GPS. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

6.4.2.2 Teléfonos móviles

El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, ya que la realización de llamadas no es dependiente de ningún terminal fijo y no requiere ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (SMS), agenda, juegos, cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3. (Baz Alonso, Ferreira Artime, Alvarez Rodrigues, & Garcia Baniello, 2009)

La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, desde el Motorola DynaTAC, el primer teléfono móvil en 1983 que pesaba 780 gramos, a los actuales más compactos y con mayores prestaciones de servicio. Además a lo largo de estos años se ha llevado a cabo el desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

Inicialmente los teléfonos móviles sólo permitían realizar llamadas de voz y enviar mensajes de texto. Conforme la tecnología fue avanzando se incluyeron nuevas aplicaciones como juegos, alarma, calculadora y acceso WAP (acceso a Internet mediante páginas web especialmente diseñadas para móviles). (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

6.4.2.3 Smartphone o teléfonos inteligentes.

Un “Smartphone” (teléfono inteligente en español) es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal. Es un elemento a medio camino entre un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional pero además incluye características cercanas a las de un ordenador personal. 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 el operador o por un tercero. (Baz Alonso, Ferreira Artime, Alvarez Rodrigues, & Garcia Baniello, 2009)

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. El completo soporte al correo electrónico parece ser una característica indispensable encontrada en todos los modelos existentes. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

Otras aplicaciones que suelen estar presentes son las cámaras integradas, la administración de contactos, el software multimedia para reproducción de música y visualización de fotos y video-clips y algunos programas de navegación así como, ocasionalmente, la habilidad de leer documentos de negocios en variedad de formatos como PDF y Microsoft Office. Una característica común a la mayoría de “Smartphone” es una lista de contactos capaz de almacenar tantos contactos como la memoria libre permita, en contraste con los teléfonos clásicos que tienen un límite para el número máximo de contactos que pueden ser almacenados. Casi todos los teléfonos inteligentes también permiten al usuario instalar programas adicionales. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

6.4.3 Sistemas Operativos para dispositivos móviles

Partiendo de la definición de sistema operativo: Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo. (GP, 2014)

Podemos deducir que el uso de uno u otro S.O determinarán las capacidades multimedia de los dispositivos, y la forma de éstas de interactuar con el usuario. Existen multitud de opciones, si bien las más extendidas son Symbian, BlackBerry OS, Windows Mobile, y recientemente iPhone OS y el sistema móvil de Google, Android, además por supuesto de los dispositivos con sistema operativo Linux. (GP, 2014)

6.4.3.1 Características básicas de cada S.O:

Symbian

Este es el sistema operativo para móviles más extendido entre “Smartphone”, y por tanto el que más aplicaciones para su sistema tiene desarrolladas. (GP, 2014)

Su principal virtud es la capacidad que tiene el sistema para adaptar e integrar todo tipo de aplicaciones. Admite la integración de aplicaciones y, como sistema operativo, ofrece las rutinas, los protocolos de comunicación, el control de archivos y los servicios para el correcto funcionamiento de estas aplicaciones. La tecnología del sistema operativo Symbian se ha diseñado teniendo en cuenta puntos clave como el poder proporcionar la

energía, memoria y gestión de entrada y salida de recursos requeridos específicamente en los dispositivos móviles. También, supone una plataforma abierta, ésta es la clave, que aúna telecomunicaciones y los estándares globales de internet. (Baz Alonso, Ferreira Artime, Alvarez Rodrigues, & Garcia Baniello, 2009)

Los usuarios de Symbian señalan como principal ventaja del sistema el hecho de que exista una amplia selección de aplicaciones disponibles para todo tipo de teléfonos móviles. Destacan también la compatibilidad con los estándares de conectividad y redes como Bluetooth, WiFi, GSM, GPRS, CDMA y WCDMA. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

Windows Mobile

Microsoft lanzó su propio Windows para móviles, antes conocido como Windows CE o Pocket PC, tiene una larga historia como segundón en el campo de los PDA u ordenadores de bolsillo, sin embargo hace pocos meses superó por primera vez al hasta entonces líder, Palm OS. (GP, 2014)

Windows Mobile es un sistema operativo escrito desde 0 y que hace uso de algunas convenciones de la interfaz de usuario del Windows de siempre. Una de las ventajas de Windows Mobile sobre sus competidores es que los programadores pueden desarrollar aplicaciones para móviles utilizando los mismos lenguajes y entornos que emplean con Windows para PC. En comparación, las aplicaciones para Symbian necesitan más esfuerzo de desarrollo, aunque también están optimizadas para cada modelo de teléfono. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

Android

Google es otro de los desarrolladores que coge algo y es capaz de convertirlo en una referencia. Android es un sistema operativo móvil basado en Linux y Java que ha sido liberado bajo la licencia Apache versión 2.

El sistema busca, nuevamente, un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones móviles y normalice las herramientas en el campo de la telefonía móvil. Al igual que ocurriera con Symbian, lo que se busca es que los programadores sólo tengan que desarrollar sus creaciones una única vez y así ésta sea compatible con diferentes terminales. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

Google ofrece una plataforma de desarrollo gratuita, flexible, económica en el desarrollo de aplicaciones y simple, diferenciada de los estándares que ofrecen Microsoft o Symbian. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2009)

IPhone OS

IPhone OS es una versión reducida de Mac OS X optimizada para los procesadores ARM. Aunque oficialmente no se puede instalar ninguna aplicación que no esté firmada por Apple ya existen formas de hacerlo, la vía oficial forma parte del iPhone Developer Program (de pago) y hay que descargar el SKD que es gratuito. IPhone dispone de un interfaz de usuario realmente interesante, la única desventaja es la cantidad de restricciones que tiene, aunque quizás Apple se dé cuenta que para triunfar mucho más es mejor liberar y dar libertad a su sistema. (GP, 2014)

BlackBerry OS

BlackBerry es un sistema operativo multitarea que está arrasando en la escena empresarial, en especial por sus servicios para correo y teclado QWERTY. BlackBerry aparece en el mercado justo en el momento en que comenzaba a demandarse un sistema operativo que permitiera utilizar de una forma fácil, cómoda y rápida los servicios de correo electrónico. Hoy en día es también proveedor de servicios de correo electrónico a dispositivos que no son BlackBerry, gracias al programa BlackBerry Connect. Así, en líneas generales, en un dispositivo BlackBerry es posible redactar, enviar y recibir todo tipo de mensajes de correo electrónico, al igual que en el programa que se utiliza en un ordenador. Además, es posible realizar y contestar las llamadas que se emitan a través de la red de telefonía móvil, lo que permite sustituir el teléfono móvil. También, como evolución lógica, los dispositivos de este fabricante permiten la navegación por internet en páginas HTML o WAP y tienen la capacidad de enviar o recibir mensajes SMS. (GP, 2014)

Por lo demás, este sistema operativo incorpora múltiples aplicaciones y programas que convierten a los dispositivos en completos organizadores de bolsillo con funciones de calendario, libreta de direcciones, bloc de notas, lista de tareas, entre otras. (GP, 2014)

6.4.4 Aplicaciones Móviles

Una aplicación móvil o App es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, es decir el mercado de aplicaciones, operadas por las compañías propietarias de los sistemas operativos móviles como Android (Play Store), iOS (APP Store), BlackBerry OS, Windows Phone (Tienda Windows), entre otros. (Martinez, 2014)

Las aplicaciones son soluciones móviles que le permiten llegar a los clientes de manera novedosa y personalizada que dinamizan la comercialización y promoción de productos y servicios, sin embargo el desarrollo de aplicaciones para dispositivos móviles requiere tener en cuenta las particularidades de estos dispositivos; Los dispositivos móviles funcionan con batería, los procesadores actualmente tienen una gran capacidad de procesamiento muy similar al PC. Los desarrollos de estas aplicaciones también tienen que considerar una gran variedad de tamaños de pantalla, datos específicos de software y configuraciones, El desarrollo de aplicaciones móviles requiere el uso de entorno de desarrollo integrado. Las aplicaciones móviles suelen ser probadas primero usando emuladores y más tarde se ponen en el mercado en periodo de prueba. (Martinez, 2014)

6. 5 Android.

6.5.1 ¿Que es Android?

Android es un sistema operativo basado en Linux (aunque con desarrollo independiente), diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó económicamente y más tarde compró en 2005. Android fue desvelado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. (Roldan, mibqyyo, 2013)

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos. (Roldan, mibqyyo, 2013)

6.5.2 Principales Características

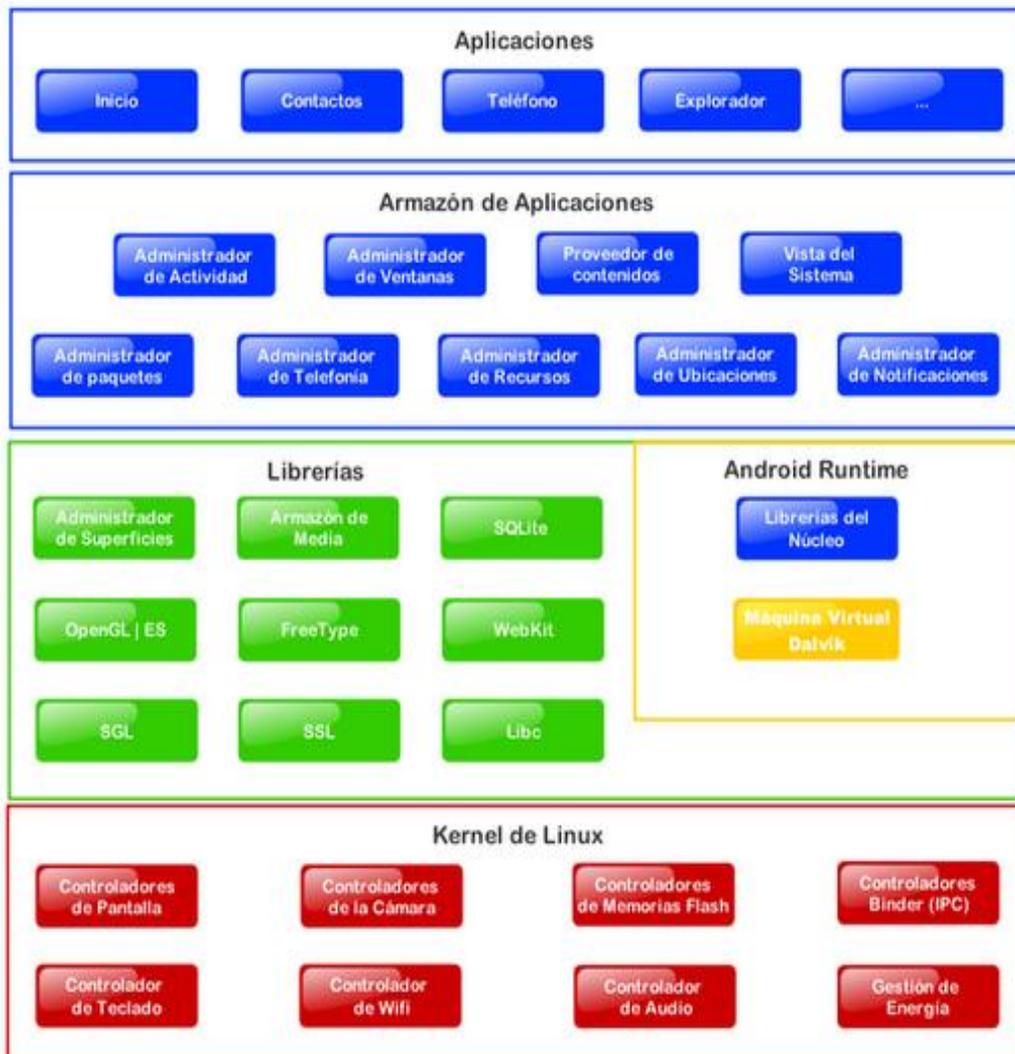
- Diseño de dispositivo: la plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0.
- Almacenamiento SQLite: una base de datos liviana.
- Conectividad Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+ y WiMAX.
- Mensajería y Navegador web.

- Soporte de Java: aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados
- Soporte multimedia: Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
- Soporte para streaming: Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML. Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
- Soporte para hardware adicional: Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
- Entorno de desarrollo: Esta característica es la que más nos interesa. Android incluye un emulador de dispositivos, herramientas para la depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo se encuentra integrado en Eclipse.
- Google Play: Un catálogo de aplicaciones gratuitas o de pago.
- Otras muchas características: Multi-táctil, Bluetooth, Video llamada, Multitarea y Tethering, que permite al teléfono o Tablets ser usado como punto de acceso inalámbrico para la conexión a Internet. (Roldan, mibqyyo, 2013)

6.5.3 Arquitectura de Android

Android es una plataforma para dispositivos móviles que contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario. (Aranaz Tudela & Campo Vazques, 2009).

En las siguientes líneas se dará una visión global por capas de cuál es la arquitectura empleada en Android. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores (Universidad Carlos III-Madrid), tal como muestra la siguiente figura:



1. **Aplicaciones:** Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.
2. **Framework de Aplicaciones:** Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel.

Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **Activity Manager:** Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.
- **Windows Manager:** Gestiona las ventanas de las aplicaciones y utiliza la librería Surface Manager.
- **Telephone Manager:** Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.).
- **Content Provider:** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta API la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- **View System:** Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, "check-boxes", tamaño de ventanas, control de las interfaces mediante teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.
- **Location Manager:** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- **Notification Manager:** Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada Intent, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.
- **XMPP Services:** Colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

3. **Librerías:** La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android.

Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- *Librería libc:* Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.

- *Librería Surface Manager:* Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.

- *OpenGL/S� y SGL:* Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/S� maneja gráficos en 3D y

permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, SGL proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.

- *Librería Media Libraries:* Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)

- *FreeType:* Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.

- *Librería SSL:* Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.

- *Librería SQLite:* Creación y gestión de bases de datos relacionales.

- *Librería WebKit:* Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

4. **Tiempo de ejecución de Android:** Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Éste lo constituyen las Core Libraries, que son librerías con multitud de clases Java y la máquina virtual Dalvik.
5. **Núcleo Linux:** Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluye un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro de este kernel de Linux embebido en el propio Android. (Universidad Carlos III-Madrid)

6.5.4 Herramientas de desarrollo

✓ **Eclipse:**

Es un entorno de desarrollo integrado de código abierto multiplataforma para crear aplicaciones. Eclipse es usado como la principal herramienta de desarrollo para java, pero también se pueden instalar plugins para otros lenguajes. (Roldan, mibqyyo, 2013)

✓ **kit de desarrollo de software (SDK)**

Un SDK (siglas en inglés de Software Development Kit) es, generalmente, un conjunto de herramientas de desarrollo de software, que permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc. (Roldan, mibqyyo, 2013)

SDK frecuentemente contienen códigos de ejemplo y notas técnicas de soporte u otra documentación, para ayudar a clarificar ciertos puntos del material de referencia primario. (Roldan, mibqyyo, 2013)

✓ **El plugin ADT**

Permite crear una AVD (Android Virtual Device) o lo que es lo mismo, un Dispositivo Virtual Android para lanzar y depurar tu aplicación. También puedes hacerlo con dispositivos externos (Tablets o Smartphone), que por conexión USB se comuniquen con tu PC y ejecutar las aplicaciones desde ellos. (Roldan, mibqyyo, 2013)

✓ **Android Developer Tools (ATD)**

Es un plugin para el IDE Eclipse, integrado en el mismo para conseguir un mayor potencial en el desarrollo e integración de aplicaciones, confirmando, en líneas generales, una mayor versatilidad para una mejor construcción de aplicaciones Android. ADT amplía las capacidades ya existentes de Eclipse, permitiendo configurar rápidamente nuevos proyectos para Android, crear una interfaz de usuario de aplicación, agregar los paquetes basados en la API Framework Android, depurar sus aplicaciones utilizando las herramientas del SDK de Android e incluso exportar con firmas las APK generadas a fin de distribuirlas adecuadamente. En resumen, integra herramientas SDK que ejecutan estas acciones. (Roldan, mibqyyo, 2013)

✓ **Dispositivo virtual de Android (AVD)**

Es una configuración del emulador que permite simular un dispositivo real mediante la definición de las opciones de software que emula una versión Android y un hardware determinado. (Roldan, mibqyyo, 2013)

Un AVD consiste en:

- Un perfil de hardware: define las características de hardware del dispositivo virtual. Por ejemplo, puedes definir si el dispositivo tiene una cámara, si utiliza un teclado QWERTY físico o un teclado para marcar, la cantidad de memoria de que dispone, etc.
- Un mapeo de una imagen del sistema: puedes definir qué versión de la plataforma Android se ejecutará en el dispositivo virtual, y elegir una versión de la imagen del sistema de empaquetado con un SDK add-on de la plataforma o estándar de Android. Puedes especificar el aspecto del emulador que deseas utilizar con la AVD, unas dimensiones de la pantalla, apariencia y un largo etcétera. (Roldan, mibqyyo, 2013)

✓ **JAVA**

Lo que se conoce genéricamente como Java abarca una serie de tecnologías basadas en varios conceptos:

- Un lenguaje de programación con el que escribir aplicaciones
- Un conjunto de librerías que facilitan el trabajo a los programadores. Las principales APIs de Sun se divide en tres bloques
- Java SE: dirigida aplicaciones de escritorio, applets.
- Java ME: enfocada en dispositivos de Consumo, con recursos limitados
- Java EE: para aplicaciones Empresariales (web, distribuidas)
- Unas Herramientas (JDK) para generar las aplicaciones a partir del código fuente.
- Una Máquina Virtual que permita ejecutar una aplicación en cualquier ordenador (JVM) (Roldan, mibqyyo, 2013)

6.5.5 Estructura de un Proyecto Android

Al crear un nuevo proyecto Android en Eclipse se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura será común a cualquier aplicación, independientemente de su tamaño y complejidad. (Salvador Gomez, 2011)

En la siguiente imagen vemos los elementos creados inicialmente para un nuevo proyecto Android:



Descripción de los elementos principales de esta estructura.

Carpeta /src/

Contiene todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc. Inicialmente, Eclipse creará por nosotros el código básico de la pantalla (Activity) principal de la aplicación, siempre bajo la estructura del paquete java definido. (Salvador Gomez, 2011)



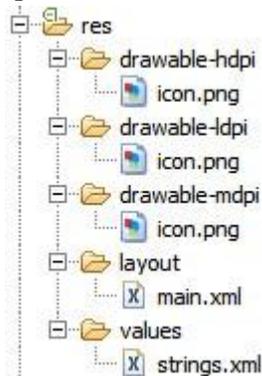
Carpeta /res/

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos de deberán distribuir entre las siguientes carpetas:

Carpeta	Descripción
/res/drawable/	<p>Contienen las imágenes de la aplicación. Para utilizar diferentes recursos dependiendo de la resolución del dispositivo se suele dividir en varias subcarpetas:</p> <ul style="list-style-type: none"> ➤ /drawable-ldpi ➤ /drawable-mdpi ➤ /drawable-hdpi
/res/layout/	<p>Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Para definir distintos layouts dependiendo de la orientación del dispositivo se puede dividir en dos subcarpetas:</p> <ul style="list-style-type: none"> ➤ /layout ➤ /layout-land

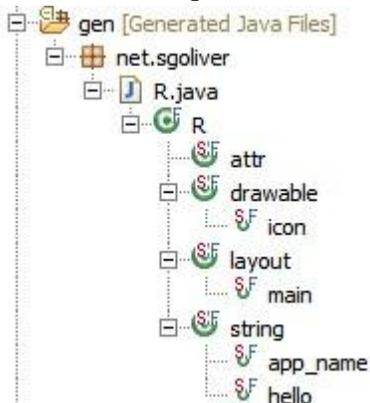
/res/anim/	Contiene la definición de las animaciones utilizadas por la aplicación.
/res/menu/	Contiene la definición de los menús de la aplicación.
/res/values/	Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc.
/res/xml/	Contiene los ficheros XML utilizados por la aplicación.
/res/raw/	Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.

Como ejemplo, para un proyecto nuevo Android, se crean los siguientes recursos para la aplicación:



Carpeta /gen/

Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de compilación de Android genera por nosotros una serie de ficheros fuente en java, dirigidos al control de los recursos de la aplicación. (Salvador Gomez, 2011)



El más importante es el que se puede observar en la imagen, el fichero R.java, y la clase R.

Esta clase R contendrá en todo momento una serie de constantes con los ID de todos los recursos de la aplicación incluidos en la carpeta /res/, de forma que podamos acceder fácilmente a estos recursos desde nuestro código a través de este dato. Así, por ejemplo, la constante R.drawable.icon contendrá el ID de la imagen “icon.png” contenida en la carpeta /res/drawable/.

Carpeta /assets/

Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.

La diferencia entre los recursos incluidos en la carpeta /res/raw/ y los incluidos en la carpeta /assets/ es que para los primeros se generará un ID en la clase R y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Usaremos uno u otro según las necesidades de nuestra aplicación. (Salvador Gomez, 2011)

Fichero AndroidManifest.xml

Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, etc.), sus componentes (pantallas, mensajes. etc.), o los permisos necesarios para su ejecución. Veremos más adelante más detalles de este fichero. (Salvador Gomez, 2011)

6.5.6 Componentes de una aplicación Android

En Java o .NET estamos acostumbrados a manejar conceptos como ventana, control, eventos o servicios como los elementos básicos en la construcción de una aplicación. Pues bien, en Android vamos a disponer de esos mismos elementos básicos aunque con un pequeño cambio en la terminología y el enfoque. Repasemos los componentes principales que pueden formar parte de una aplicación Android [Por claridad, y para evitar confusiones al consultar documentación en inglés, intentaré traducir lo menos posible los nombres originales de los componentes]. (Salvador Gomez, 2011)

Activity

Las actividades (activities) representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual. (Salvador Gomez, 2011)

View

Los objetos view son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los controles de Java o .NET. De inicio, Android pone a nuestra disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados. (Salvador Gomez, 2011)

Services

Los servicios son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son exactamente iguales a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (p.ej. activities) si se necesita en algún momento la interacción con del usuario. (Salvador Gomez, 2011)

Content Provider

Un content provider es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido. (Salvador Gomez, 2011)

Broadcast Receiver

Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: “Batería baja”, “SMS recibido”, “Tarjeta SD insertada”...) o por otras aplicaciones (cualquier aplicación puede generar mensajes (intents, en terminología Android) Broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo). (Salvador Gomez, 2011)

Widget

Los widgets son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (home screen) del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal. (Salvador Gomez, 2011)

Intent

Un intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje Broadcast, iniciar otra aplicación, etc.

6.5.7 Versiones de Android y Niveles de API

Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre comercial. El nivel de API corresponde a números enteros comenzando desde 1. Para los nombres comerciales se han elegido postres en orden alfabético Cupcake (v1.5), Donut (v1.6), Éclair (v2.0), Froyo (v2.2), Gingerbread (v2.3), etc. Las dos primeras versiones, que hubieran correspondido a las letras A y B, no recibieron nombre. (Nuñez, 2013)

6.5.7.1 Características de las principales versiones.

Éclair-Android 2.0/2.1 Nivel de API 5 (Octubre 2009)

- Velocidad de Hardware optimizada.
- Interfaz de usuario renovada.
- Nuevas listas de contacto.
- Mejoras en google maps.
- Zoon digital.
- Fondos de pantallas animados.
- Soporte integrado de flash.
- Teclado Virtual mejorado, entre otros. (Nuñez, 2013)

Froyo-Android 2.2 Nivel de API 8 (mayo 2010)

- Optimización general del sistema Android, la memoria y el rendimiento.
- Mejora en la velocidad de Aplicaciones.
- Lanzador de Aplicaciones mejorado con accesos directos a las aplicaciones de teléfono.
- Actualización del market con actualizaciones automáticas.

- Permite desactivar el tráfico de datos.
- Compartir contactos por Bluetooth.
- 2.2.1, 2.2.2, 2.2.3, actualizadas en el 2010 con arreglo de errores. (Nuñez, 2013)

Gingerbread-Android 2.3 Nivel de API 9 (diciembre 2010)

- Soporte para dispositivos móviles.
- Actualización del diseño de la interfaz del usuario.
- Soporte para pantallas extra grandes y resoluciones mayores.
- Soporte mejorado para desarrollo de código nativo.
- Recolección de elementos concurrentes para un mayor rendimiento.
- Un administrador de descargas para archivos grandes.
- Soporte nativo para múltiples cámaras, Hubo mejoras al sistema y arreglo en fallas del 2.2.3 al 2.3.7. (Nuñez, 2013)

Honeycomb- Android 3.0 Nivel de API 11 (febrero 2011)

- Mejor soporte para Tablets, Escritorio 3D con widgets rediseñados.
- Mejoras en el navegador web.
- Sistemas multitarea mejorado.
- Soporte para video chat mediante Google Talk.
- Añade soporte para una gran variedad de periféricos y accesorios con conexión USB, teclado, ratones, hubs, dispositivos de juegos y cámara digitales. (Nuñez, 2013)

Ice Cream Sandwich- Android 4.0 Nivel de API 14 (octubre 2011)

- Unifica las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo.
- Interfaz limpia y moderna.
- Multitarea mejorada.
- Gestor de tráfico de datos de internet.
- Corrector de texto, rediseñado y mejorado. (Nuñez, 2013)

Jelly Bean- Android 4.1 Nivel de API 16 (julio 2012)

- Mejora la funcionalidad y rendimiento de la interfaz de usuario.
- Notificaciones mejoradas, con acceso más rápido a más información.
- Google chrome se convierte en navegador por defecto.
- Cambios en la interfaz como nueva barra de búsqueda. (Nuñez, 2013)

6.6 Java ME

6.6.1 Java

Java es un lenguaje de programación orientado a objetos creado por Sun Microsystems. Su principal aliciente, y la característica que lo convirtió en uno de los lenguajes más populares hasta hoy, fue la posibilidad de crear aplicaciones independientes a la plataforma donde van a ser ejecutadas, siguiendo el axioma “*write once, run anywhere*” (Universidad Carlos III-MAdrid)

Java ofrece la oportunidad de crear aplicaciones tanto de escritorio como empresariales, incluyendo por supuesto a los dispositivos móviles. Esta portabilidad viene dada por la máquina virtual de Java o **JVM** (*Java Virtual Machine*). Al compilar un fichero fuente en Java no se crea un binario directamente ejecutable, sino un código intermedio llamado **bytecode** en ficheros cuya extensión es **.class**. Para cada hardware debe haber una JVM específica, ya sea un teléfono móvil, un ordenador con Windows, o un microondas. Cada máquina virtual conoce el conjunto de instrucciones de la plataforma sobre la que está instalada, y puede traducir el bytecode, común para todas las máquinas, al código nativo que es capaz de entender el hardware en cuestión de dicha plataforma. (Universidad Carlos III-MAdrid)

6.6.2 Ediciones de Java 2

La revolución tecnológica acaecida en los últimos años propició que los responsables de Java ofrecieran soluciones personalizadas a cada ámbito tecnológico. Sun decidió crear una edición distinta de Java según las necesidades del entorno y la tecnología utilizada (Universidad Carlos III-MAdrid):

- Java Enterprise Edition (Java EE), orientada al entorno empresarial.
- Java Standard Edition (Java SE), orientada al desarrollo con independencia de la plataforma.

- Java Micro Edition (Java ME), orientada a dispositivos con capacidades restringidas.
- Java Card, orientada a tarjetas inteligentes ("smart cards").

Todas las ediciones comparten un conjunto más o menos amplio de las API básicas de Java, agrupadas principalmente en los paquetes `java.lang` y `java.io`. A día de hoy, Java EE representa un súper conjunto de Java SE, pues contiene toda la funcionalidad de éste y más características, así como Java ME es un subconjunto de Java SE (excepto por el paquete `javax.microedition`). Por lo tanto, Java Micro Edition es una versión muy específica del lenguaje Java, creado para desarrollar, instalar y ejecutar software escrito en Java en aparatos electrónicos de baja capacidad, como electrodomésticos, PDA, teléfonos móviles y ordenadores de bolsillo. (Universidad Carlos III-Madrid)

6.6.3 Arquitectura Java ME

Una aplicación Java ME típica está formada por un archivo JAR, que es el que contiene a la aplicación en sí, y un archivo opcional JAD (Java Archive Descriptor) que contiene diversa información sobre la aplicación. Antes de desarrollar una aplicación en Java ME, es necesario tomar una serie de decisiones según el dispositivo a utilizar y las necesidades requeridas. No todos los desarrollos realizados en Java ME utilizan los mismos componentes. En concreto, una aplicación en Java ME se desarrolla a partir de una combinación de:

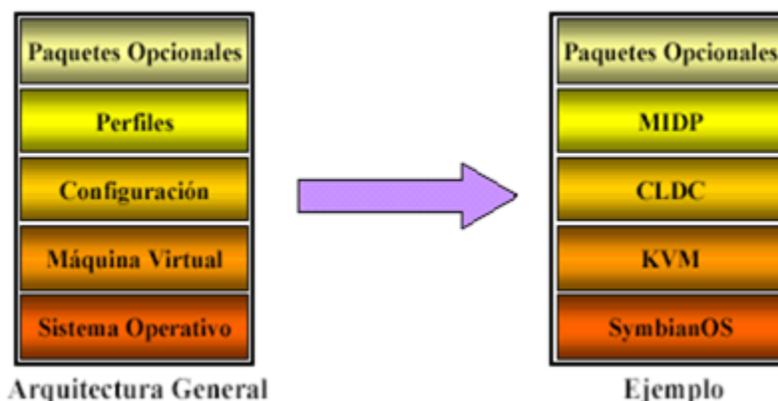
- Máquina virtual: Existen dos máquinas virtuales de Java ME con diferentes requisitos, cada una pensada para tipos distintos de pequeños dispositivos: KVM (Kilobyte Virtual Machine) y CVM (Compact Virtual Machine).
- Configuración: Una configuración consiste en un conjunto de clases básicas destinadas a conformar el corazón de la aplicación. En concreto, dentro de Java ME existe dos configuraciones: CLDC (Connected Limited Device Configuration) y CDC (Connected Device Configuration).
- Perfil: Bibliotecas de clases específicas, orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos (Universidad Carlos III-Madrid).

Cada una de las dos configuraciones mencionadas requiere en realidad de una determinada máquina virtual. De esta forma, si se escoge la configuración CLDC, será necesario utilizar la máquina virtual denominada KVM; si, por el contrario, se decide

utilizar la configuración CDC, la máquina virtual necesaria es la conocida como CVM (Universidad Carlos III-Madrid).

Con la elección de perfiles, se da una situación similar. Existen unos perfiles que se utilizan sobre la configuración CDC (Foundation Profile, Personal Profile y Personal Basis Specification) y otros que lo hacen sobre CLDC (Mobile Information Device Profile, conocido como MIDP, e Information Module Profile). (Universidad Carlos III-Madrid)

Las aplicaciones en Java ME que se realizan utilizando el perfil MIDP reciben el nombre de MIDlets. Se dice así que un MIDlet es una "aplicación Java ME realizada con el perfil MIDP", sobre la configuración CLDC, y usando la máquina virtual KVM. Desde un punto de vista práctico, MIDP es el único perfil actualmente disponible. (Universidad Carlos III-Madrid)



6.6.4 Librerías

Al desarrollar una aplicación en Java ME bajo el perfil MIDP, el programador puede hacer uso, entre otras, de las siguientes bibliotecas:

Nombre	Descripción
java.io	Operaciones de E/S básicas
java.lang	Operaciones de la máquina virtual
java.util	Utilidades estándar
javax.microedition.midlet	Marco de ejecución para las aplicaciones
javax.microedition.lcdui	Interfaces de usuario
javax.microedition.rms	Almacenamiento persistente en el dispositivo
javax.microedition.io	Conexión genérica

A través de la configuración CLDC, se obtienen un conjunto de clases heredadas de la edición Java SE. Aproximadamente, unas 37 clases derivadas de los paquetes java.lang, java.util y java.io. Cada clase es un subconjunto de la correspondiente en Java SE. (Universidad Carlos III-MAdrid)

Los paquetes java.io y java.net, son los encargados en Java SE de las operaciones de E/S. Debido a las limitaciones de CLDC, no fue posible incluir todas y cada una de las clases de estos paquetes. Tampoco se incluyeron la clase java.net, encargada de las comunicaciones TCP/IP. En su lugar, MIDP añade el paquete javax.microedition.io que suple de forma genérica las necesidades en cuestiones de comunicación. Además de los ya mencionados, MIDP incluye también un paquete que posibilita la creación de interfaces de usuario: javax.microedition.lcdui. Las clases contenidas en este paquete pueden dividirse en dos grandes grupos:

-> **Elementos de interfaz de usuario de alto nivel:** Esta interfaz usa componentes tales como cajas de texto, alertas, listas, etc. Estos elementos son implementados por cada dispositivo y la finalidad de usar las APIs de alto nivel es su portabilidad.

-> **Elementos de interfaz de usuario de bajo nivel:** Se tiene un control total sobre los recursos, pero se puede perder la portabilidad ya que dependerá mucho del dispositivo destino. Generalmente, se utiliza en juegos (Universidad Carlos III-MAdrid).

El paquete javax.microedition.rms proporciona un mecanismo a los MIDlets para poder almacenar información de forma persistente en el dispositivo, utilizando para ello una base de datos basada en registros llamada *Record Management System o RMS*. Los registros se convierten así en la unidad básica de información en un MIDlet, permitiendo al dispositivo crear, leer, modificar y borrar cualquier registro de datos (Universidad Carlos III-MAdrid).

6.6.5 Comunicaciones en MIDlets

El paquete javax.microedition.io contiene las clases que dan soporte a las comunicaciones en una aplicación MIDP. De hecho, este paquete pretende sustituir, de una forma más breve y concisa, al paquete java.net de Java SE. Sin embargo, en una comunicación el MIDlet no utiliza únicamente este paquete, sino que también usará java.io, añadido por la configuración CLDC. Mientras que el paquete javax.microedition.io tiene como misión crear y manejar diferentes conexiones de red (por HTTP, datagramas, sockets), el paquete java.io lo que posibilita a través de su declaración de clases es poder leer y escribir en estas conexiones de red. Debido a las limitaciones ya conocidas de un

dispositivo basado en MIDP, es del todo imposible contener en él la implementación de cada protocolo de comunicación existente.

Lo que proporciona `javax.microedition.io` es una única clase, llamada `Connector`, que esconde todos los detalles de la conexión. Dicho en otras palabras, una misma clase `Connector` sirve tanto para leer archivos, como abrir una conexión por HTTP o mediante sockets. (Universidad Carlos III-MAdrid)

6.7 Análisis y Diseño de Sistemas

El Análisis y Diseño de Sistemas permite examinar la situación de una organización con el propósito de mejorarla con métodos y procedimientos más adecuados. Uno de los cambios más comunes que suelen darse en las instituciones después de hacer un minucioso análisis es el de pasar de un sistema manual a un sistema computarizado muchas de las actividades cotidianas que se realizan.

El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: el análisis de sistemas y el diseño de sistemas.

El *análisis de sistemas*, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

El *diseño de sistemas* es el proceso de planificar, reemplazar o complementar un sistema organizacional existente, pero antes de llevar a cabo esta planeación es necesario comprender, en su totalidad, el viejo sistema y determinar la mejor forma en que se pueden utilizar las computadoras para hacer la operación más eficiente.

6.7.1. Modelos de Desarrollo de Sistemas

6.7.1.1 ¿Qué es un modelo de desarrollo?

Un modelo de desarrollo es una representación abstracta de un proceso de software, cada modelo representa el proceso de desarrollo de software de una manera en particular. A pesar de estar definidos claramente, no representan necesariamente la realidad de cómo se debe desarrollar el software, sino que establece un enfoque común. Un modelo puede ser modificado y adaptado de acuerdo a las necesidades del software en desarrollo. (Parra, 2013)

En forma general podemos clasificar los modelos de desarrollo en 3 grupos:

1. El modelo en cascada. Considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera. (Parra, 2013)

2. Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades. (Parra, 2013)

1. Ingeniería del software basada en componentes. Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero. (Parra, 2013)

Aunque existen muchos tipos de modelos de desarrollo, de forma genérica la mayoría está clasificada en una de estas 3 categorías, y estos a pesar de ser diferentes a veces son usados de manera simultáneamente especialmente en sistemas grandes.

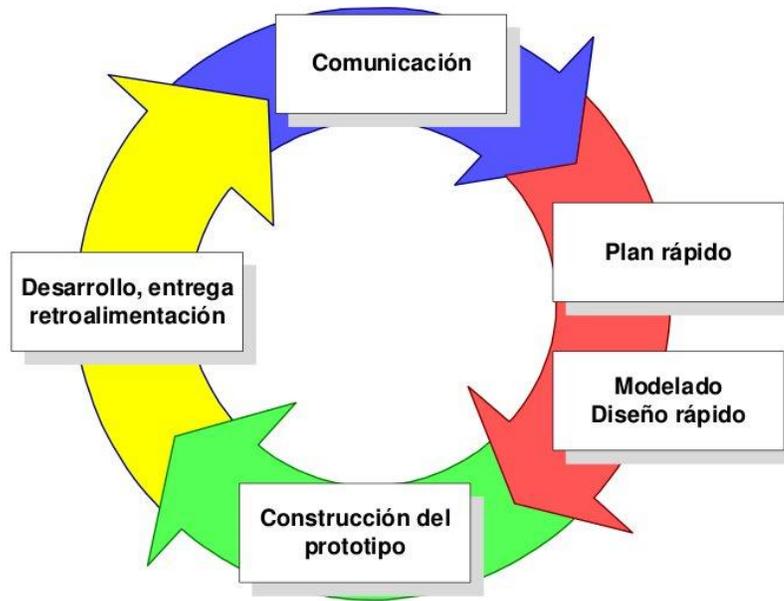
6.7.1.2 Modelos Desarrollo Evolutivos

El desarrollo evolutivo consta del desarrollo de una versión inicial que luego de exponerse se va refinando de acuerdo de los comentarios o nuevos requerimientos por parte del cliente o del usuario final. Las fases de especificación, desarrollo y validación se entrelazan en vez de separarse.

Los modelos evolutivos son iterativos; los caracteriza la forma en que permiten que los ingenieros de software desarrollen versiones cada vez más completas del software.

- **Construcción de Prototipos**

En este modelo de ciclo de vida se desarrolla el concepto del sistema a medida que avanza el proyecto. Se inicia desarrollando los aspectos más visibles del sistema. Se presenta al cliente la parte ya desarrollada del proyecto y se continúa el desarrollo del prototipo con base en la realimentación que se recibe del cliente. El ciclo continúa hasta que el prototipo se convierte en el producto final de ingeniería. La gráfica del presente modelo sería la siguiente:



Ventajas:

- Ideal para proyectos cuyos requerimientos cambian con rapidez.
- Cuando el cliente no puede especificar el conjunto total de los requerimientos.
- Cuando no se logra identificar de forma apropiada el área de aplicación.
- Cuando los desarrolladores no están seguros de la arquitectura o los algoritmos adecuados a utilizar.

Desventajas

- Existe una imposibilidad de conocer al inicio del proyecto lo que se tardará en crear un producto aceptable.
- Esta aproximación puede convertirse fácilmente en una excusa para realizar el desarrollo con el modelo de codificar y corregir

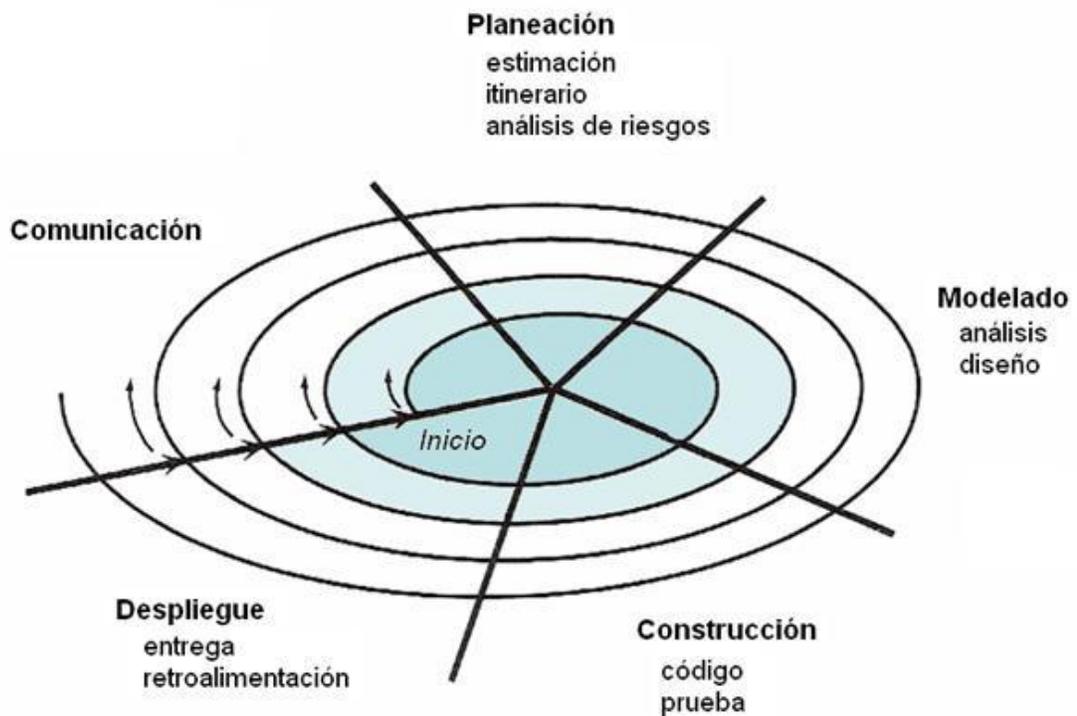
- **Modelo en espiral**

El Modelo de desarrollo en espiral es un generador del modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de software concurrente y con múltiples usuarios. Tiene dos características distintivas principales. Una de ellas es un enfoque cíclico para el crecimiento incremental del grado de riesgo. La otra

es un conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones de sistema que sean factibles y mutuamente satisfactorias.

Cuando se aplica el modelo en espiral, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, la entrega tal vez sea un documento del modelo de prototipo. Durante las últimas iteraciones se producen versiones cada vez más completas del sistema desarrollado.

Un proceso en espiral se divide en un conjunto de actividades del marco de trabajo que define el equipo de ingeniería del software. Para propósitos ilustrativos se aprovechan las actividades del marco de trabajo representa un segmento de la ruta en espiral que se representa en la figura. Cuando comienza este proceso evolutivo el equipo de desarrollo realiza actividades implicadas en un circuito alrededor de la espiral que tiene una dirección en el sentido del movimiento de las manecillas del reloj, y que se inicia desde el centro.

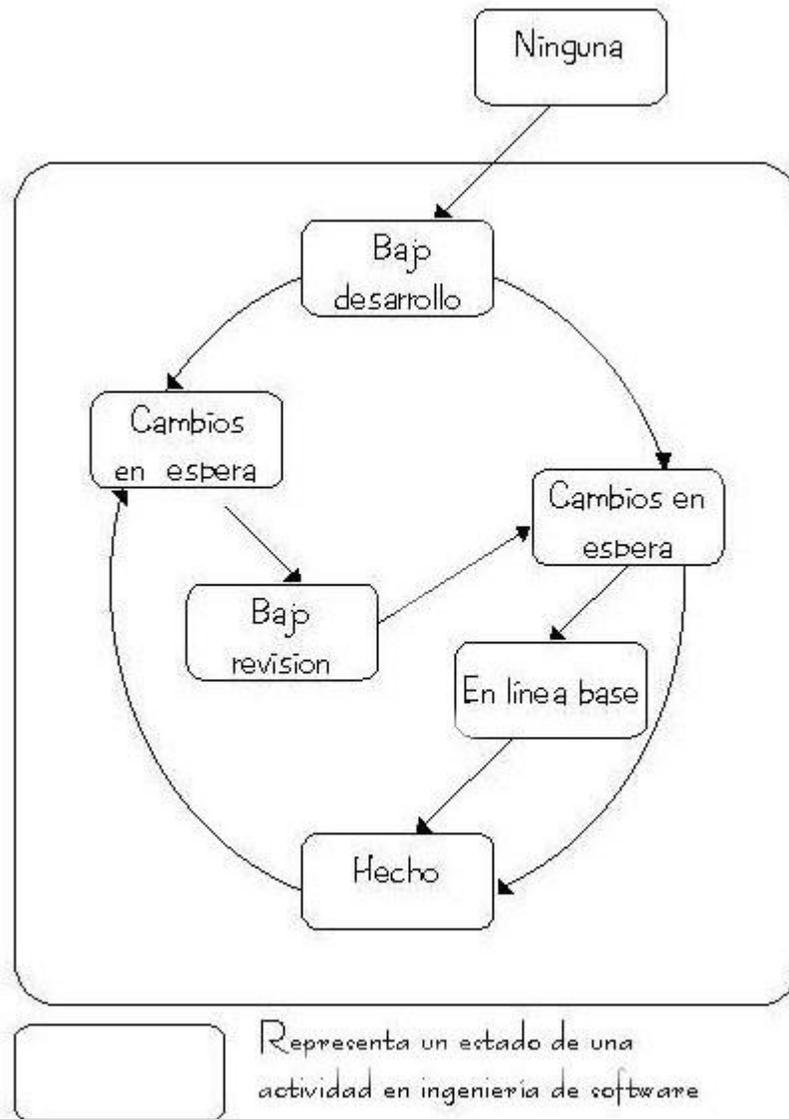


- **Modelo de desarrollo concurrente**

El modelo de desarrollo concurrente, llamado algunas veces ingeniería concurrente, se representa en forma esquemática como una serie de actividades del marco de trabajo, acciones y tareas de la ingeniería del software y sus estados asociados. Por ejemplo, la

actividad de modelado, definida para el modelo espiral, se lleva a cabo al invocar las acciones siguientes: construcción de prototipos o modelado y especificación del análisis y diseño.

En la figura 3.6 se proporciona un esquema de una tarea de ingeniería de software relacionada con la actividad de modelado de proceso concurrente. La actividad – modelado- puede estar en algunos de los estados destacados mencionados antes en cualquier momento dado. De forma similar, otras actividades o tareas (por ejemplo, la comunicación y la construcción) se representan de una manera análoga. Todas las actividades existen de forma concurrente, pero se encuentran en diferentes estados. Por ejemplo, al principio del proyecto la actividad de comunicación (no se muestra en la figura) ha completado su primera iteración y existe en el estado de en espera de cambios. La actividad de modelado que existió en el estado ninguno mientras se realizaba la comunicación inicial, ahora realiza una transición al estado en desarrollo. Sin embargo, si el cliente indica cambios en los requisitos, la actividad de modelado se mueve del estado en desarrollo al estado de en espera de cambios.



Modelo evolutivo: desarrollo concurrente de Davis Sitaram

El modelo de proceso concurrente define una serie de eventos que dispararan transiciones de estado a estado para cada una de las actividades, acciones o tareas de la ingeniería del software.

6.7.1.3. Prototipo.

✓ Definición

La elaboración de prototipos es una técnica de recopilación de información útil para complementar el ciclo de vida de desarrollo de un sistema tradicional. Cuando el analista de sistemas usa prototipos está buscando reacciones, sugerencias, innovaciones y planes de revisión del usuario para hacer mejoras al prototipo y por lo tanto, modificar los planes del sistema con un mínimo de gastos y trastornos. Los sistemas que apoyan la toma de decisiones semiestructuradas (tal como lo hacen los sistemas de apoyo a decisiones) son buenos candidatos para la elaboración de prototipos. (Kendall & Kendall, 2005)

✓ Tipos de prototipos

Prototipo parchado:

El primer tipo en la elaboración de prototipos tiene que ver con la construcción de un sistema que trabaja, pero que está parchado.

Un ejemplo de un prototipo en sistemas de información es un modelo operable que tiene todas las características necesarias, pero que es ineficiente. (Kendall & Kendall, 2005)

Prototipo no operacional:

La segunda concepción de un prototipo es la de un modelo a escala no funcional para objeto de probar determinados aspectos del diseño. (Kendall & Kendall, 2005)

Prototipo primero de una serie:

Una tercera concepción de la elaboración de prototipos involucra la creación de un primer modelo a escala completa de un sistema, llamado a veces un piloto. (Kendall & Kendall, 2005)

Prototipo de características seleccionadas:

Una cuarta concepción de la elaboración de prototipos se refiere a la construcción de un modelo operacional que incluye algunas, pero no todas, de las características que tendrá el sistema final. (Kendall & Kendall, 2005)

Prototipo Desechable:

Paradigma de creación cerrado, sirve únicamente como demostración de los requisitos, después se redacta la especificación del sistema y se desecha el prototipo; la

aplicación se desarrolla siguiendo un paradigma diferente. (Ingeniería del Software, 2009)

Prototipo Evolutivo:

Paradigma de creación abierta, comienza con un sistema relativamente simple que implementa los requisitos más importantes o mejor conocidos, se aumenta o cambia en cuanto se descubren nuevos requisitos y finalmente, se convierte en el sistema requerido. (Ingeniería del Software, 2009)

✓ **Lineamientos para el desarrollo de un prototipo**

1. Trabajo en módulos manejables.
2. Construir el prototipo rápidamente
3. Modificar el prototipo en iteraciones sucesivas
4. Enfatizar la interfaz del usuario. (Kendall & Kendall, 2005)

✓ **Desventajas de los prototipos**

Una desventaja de los prototipos es que el manejo del proceso de elaboración del prototipo es difícil, debido a la rapidez del proceso y a sus muchas iteraciones.

Una segunda desventaja es que puede haber presiones para que sea puesto en servicio un prototipo incompleto, como si fuera un sistema completo. (Kendall & Kendall, 2005)

✓ **Ventajas de los prototipos**

1. El potencial para cambiar el sistema en etapas tempranas de su desarrollo.
2. La oportunidad de detener el desarrollo de un sistema que no es funcional.
3. La posibilidad de desarrollar un sistema que satisfaga en mejor forma las necesidades y expectativas de los usuarios. (Kendall & Kendall, 2005)

✓ **Papel del usuario en los prototipos**

El primer interés de los usuarios debe ser interactuar con el prototipo mediante experimentación. Los analistas de sistemas deben trabajar sistemáticamente para obtener y evaluar las reacciones de los usuarios ante el prototipo, y luego trabajar para incorporar las sugerencias e innovaciones de los usuarios que valgan la pena en las modificaciones subsecuentes. (Kendall & Kendall, 2005)

6.7.2. Metodología de Investigación.

La metodología de la investigación es el conjunto de métodos y técnicas que se utilizan durante el proceso de una investigación. (Tipos de investigación, 2008)

Existen varias maneras para clasificar los tipos de investigación científica entre las cuales tenemos:

Según el grado de abstracción:

- **Pura** (o básica): Investigación de nuevos conocimientos, con el propósito de aumentar la teoría, despreocupándose de las aplicaciones prácticas que puedan derivarse. (Tipos de investigación, 2008)
- **Aplicada**: Está orientada a la resolución de problemas prácticos, con un margen de generalización limitado. Su propósito de realizar aportes al conocimiento científico es secundario. (Tipos de investigación, 2008)

Según la naturaleza de los datos:

- **Cuantitativa**: Admite la posibilidad de aplicar las ciencias sociales el método de investigación de las Ciencias Físico-Naturales. El objeto de estudio es “externo” al sujeto que lo investiga tratando de lograr la máxima objetividad. Intenta identificar leyes generales referidas a grupos de sujeto o hechos. Sus instrumentos suelen recoger datos cuantitativos los cuales también incluyen la medición sistemática, y se emplea el análisis estadístico como característica resaltante. (Tipos de investigación, 2008)
- **Cualitativa**: Es una investigación que se basa en el análisis subjetivo e individual, esto la hace una investigación interpretativa, referida a lo particular. (Tipos de investigación, 2008)

Según la temporalización.

- **Métodos Transversales**: Se realiza en un lapso de tiempo corto, se recolectan datos en un solo momento, su propósito es describir variables y analizar su incidencia o interrelación en un momento dado. Es como tomar una instantánea de un evento. (Tipos de investigación, 2008)

- **Métodos Longitudinales:** El estudio se hace en un tiempo prolongado viendo la evolución del evento bajo estudio, los datos se recolectan a través del tiempo en puntos o periodos específicos. Es como una película de un evento. (Tipos de investigacion, 2008)

Según las fuentes.

- **Investigación Bibliográfica:** Es la revisión bibliográfica de tema para conocer el estado de la cuestión. La búsqueda, recopilación, organización, valoración, crítica e información bibliográfica sobre un tema específico tiene un valor, pues evita la dispersión de publicaciones o permite la visión panorámica de un problema. (Tipos de investigacion, 2008)
- **Investigación Metodológica:** Indaga sobre los aspectos teóricos y aplicados de medición, recolección y análisis de datos o de cualquier aspecto metodológico. (Tipos de investigacion, 2008)
- **Investigación Empírica:** Se basa en observación y experimentación, puede emplear metodología cualitativa y cuantitativa, razonamiento hipotético-deductivo, ser de campo o laboratorio y se pueden emplear métodos transversales o longitudinales, entre otros. (Tipos de investigacion, 2008)

6.7.3. Métodos de recolección de datos

Para la recolección de datos primarios en una investigación científica se procede básicamente por observación, por encuestas o entrevistas a los sujetos de estudio y por experimentación. (Kendall & Kendall, 2005)

Encuesta:

Constituye el término medio entre la observación y la experimentación. En ella se pueden registrar situaciones que pueden ser observadas y en ausencia de poder recrear un experimento se cuestiona a la persona participante sobre ello. Por ello, se dice que la encuesta es un método descriptivo con el que se pueden detectar ideas, necesidades, preferencias, hábitos de uso, etc. (Kendall & Kendall, 2005)

La Entrevista.

La entrevista es el instrumento más importante de la investigación, junto con la construcción del cuestionario. En una entrevista además de obtener los resultados subjetivos del encuestado acerca de las preguntas del cuestionario, se puede observar la

realidad circundante, anotando el encuestador además de las respuestas tal cual salen de la boca del entrevistado, los aspectos que considere oportunos a lo largo de la entrevista. (Kendall & Kendall, 2005)

La entrevista, a diferencia del cuestionario que se contesta por escrito por el encuestado, tiene la particularidad de ser más concreta, pues las preguntas presentadas de forma contundente por el encuestador, no dejan ambigüedades, es personal y no anónima, es directa por qué no deja al encuestado consultar las respuestas. Es un método cómodo para obtener datos referentes a la población, facilitados por individuos y que nos sirven para conocer la realidad social. Estos datos podrían observarse directamente a través de la observación pero serían subjetivos de los investigadores, resultando más costosa su obtención. (Kendall & Kendall, 2005)

6.7.4. Base de Datos

¿Qué es una base de datos?

Es un conjunto de datos almacenados en un dispositivo de almacenamiento masivo, como por ejemplo:

- a) Discos Duros
- b) CD-ROM
- c) Memorias USB

Los cuales se encuentran disponibles en forma simultánea a un número de usuarios autorizados y en un tiempo pertinente.

6.7.4.1. Base de Datos Relacional

Una parte fundamental para lograr la eficiencia de un sistema consiste en el uso de las bases de datos, que según Kendall & Kendall persigue los siguientes objetivos:

1. Asegurarse de que la base de datos pueda ser compartida entre los usuarios de una diversidad de aplicaciones.
2. Mantener datos que sean precisos y consistentes.
3. Asegurarse de que todos los datos requeridos para las aplicaciones actuales y futuras estén fácilmente disponibles.
4. Permitir que la base de datos evolucione y que las necesidades de los usuarios crezcan.
5. Permitir que los usuarios construyan su vista personal de los datos sin preocuparse de la forma en que estén físicamente guardados los datos.

El compartir los datos significa que estos deben estar guardados solamente una vez. Esto, a su vez, logra la integridad de los datos, debido a que los cambios a los mismos se logran mucho más fácil y confiablemente si los datos aparecen una sola vez, en lugar de estar en muchos archivos diferentes.

Una base de datos puede evolucionar conforme cambian las necesidades de los usuarios y las aplicaciones. En nuestro caso sería óptimo usar este tipo de estructura.

Muchos sistemas de información, ya sean implantados en sistemas de cómputos grandes o pequeños, interactúan con bases de datos que abarcan varias aplicaciones. Dada la importancia que tienen las bases de datos, su diseño es establecido y vigilado por un administrador de base de datos.

➤ **Estructuras de datos relacionales**

Una estructura de datos relacional consiste en una o más tablas de dos dimensiones a las que se les llama relaciones. Los renglones (tuplas) de la tabla representan los registros y las columnas contienen atributos.

El mantener las tablas en una estructura relacional es, por lo general, bastante simple en comparación con el mantenimiento de una estructura jerárquica o de red.

Una de las principales ventajas de la estructura relacional es que las preguntas **AD HOC** son manejadas eficientemente.

Un archivo es llamado una relación, un registro es llamado, por lo general, una tupla y el conjunto de valores de atributo es llamado dominio.

➤ **Diagrama Entidad – Relación**

La estructura lógica general de una base de datos puede expresarse en forma gráfica por medio de un *diagrama ER* que se integra con los siguientes componentes:

- **Rectángulos:** representan conjuntos de entidades.
- **Elipses:** representan atributos.
- **Rombos:** representa conjuntos de relaciones.
- **Líneas:** conectan los atributos a los conjuntos de entidades, y los conjuntos de entidades a los conjuntos de relaciones.

Cada componente se etiqueta con su nombre correspondiente. Para distinguir las cardinalidades de las relaciones se dibuja líneas con y sin dirección.

Un conjunto de entidades débiles se indica en los diagramas ER por medio de un rectángulo con doble línea la relación que la conecta al conjunto de entidades fuerte en el que se forma su llave primaria se señala mediante líneas gruesas.

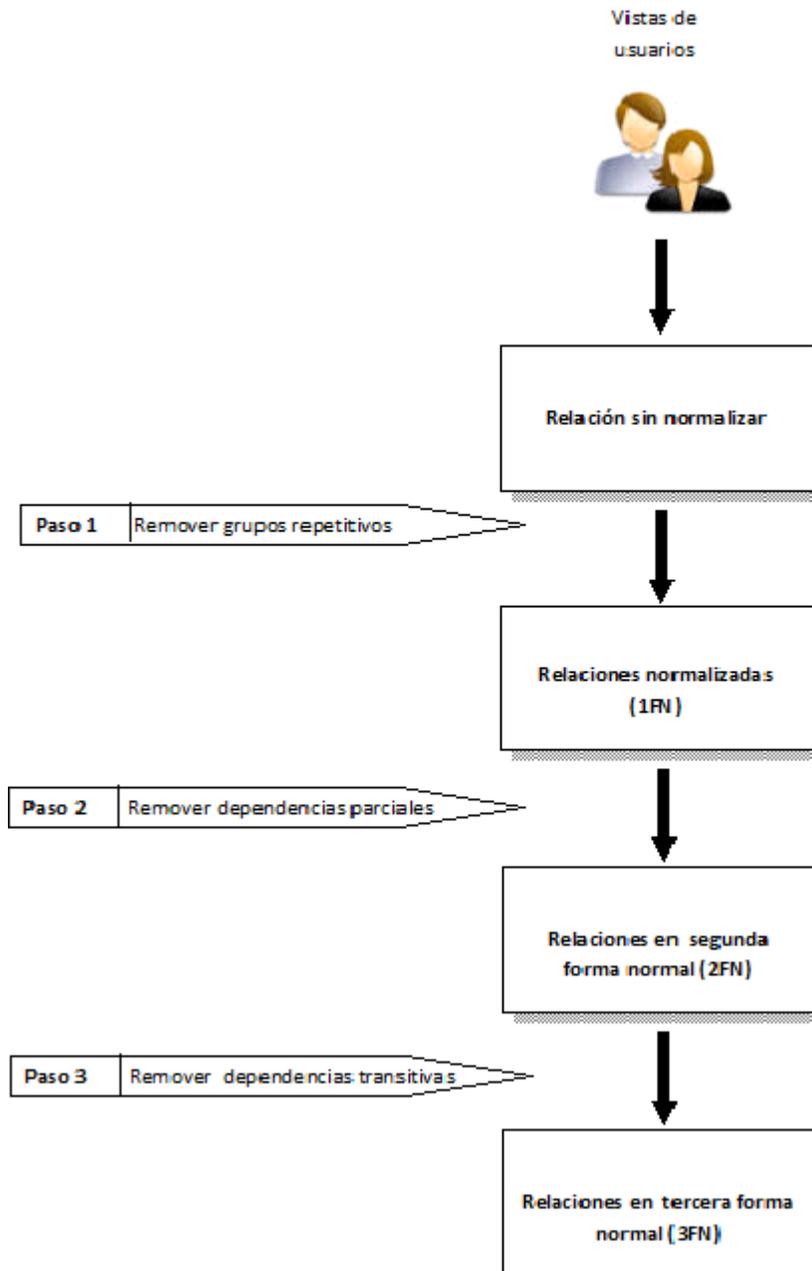
6.7.4.2. Normalización de Base de Datos

Definición

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables. Además de ser más simples y estables, las estructuras de datos normalizadas son más fáciles de mantener que otras estructuras de datos.

Los tres pasos de la normalización

Ya sea que empiece con una vista de usuarios o un almacén de datos desarrollado para un diccionario de datos, el analista normaliza una estructura de datos en tres pasos, como se muestra en la figura. Cada paso involucra un procedimiento importante, el cual simplifica la estructura de datos.



La relación derivada de la vista de usuario o del almacén de datos probablemente no estará normalizada. El primer paso del proceso incluye quitar todos los grupos repetitivos e identificar la clave primaria. Para ello, la relación se debe dividir en dos o más relaciones. A estas alturas, las relaciones ya podrían ser la tercera forma normal, pero probablemente se necesitarán más pasos para transformar las relaciones a la tercera forma normal.

El segundo paso asegura que todos los atributos sin clave son totalmente dependientes de la clave primaria. Todas las dependencias parciales se remueven y se ponen en otra relación.

El tercer paso remueve cualesquiera dependencias transitivas. Una dependencia transitiva es aquella en la que los atributos sin clave son dependientes de otros atributos sin clave.

6.7.5. Diccionario de Datos.

Concepto

Es un repositorio, manual o automatizado que contiene definiciones de todos los objetos de datos consumidos y producidos por un sistema de información.

Por lo general, un diccionario de datos es a su vez una base de datos y por lo tanto se puede almacenar y actualizar por medio de un sistema computacional (Kendall & Kendall, 2005).

Así, el diccionario de datos tiene una doble misión de organización y control de los datos. Este es el núcleo de cualquier esquema conceptual.

Objetivos de un diccionario de datos

- Permitir la administración de la documentación de los datos y a los usuarios entender lo que son los datos y que significan las definiciones y descripciones.
- Ayudar a controlar mejor los recursos de datos, ya que con un diccionario de datos la información sobre ellos permanece en un solo lugar.

Un diccionario de datos *independiente* es aquel creado, por ejemplo, por compañías consultoras que emplean un DBMS (Sistema Administrador de Bases de Datos) para administrar su meta-dato o brindan sus propios medios para almacenar y recuperar información. Un diccionario de este tipo con frecuencia sirve de interfaz a varios DBMS. Un diccionario de datos *dependiente* forma parte del DBMS y lo emplea para administrar sus datos.

En cuanto a los beneficios de usar un diccionario de datos, se pueden mencionar los siguientes:

- Lograr homogeneidad en la representación de los datos.
- Lograr concordancia en las definiciones de los datos.

- Cumplir con los estándares de los datos.
- Eliminar la redundancia de los datos.
- Reducir el esfuerzo de análisis de los datos.

6.7.6. SQL (Structured Query Language)

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Componentes del SQL:

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Comandos:

Existen tres tipos de comandos SQL:

- Los **DDL (Data Definition Language)** que permiten crear y definir nuevas bases de datos, campos e índices.
- Los **DML (Data Manipulation Language)** que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
- Los **DCL (Data Control Language)** que se encargan de definir los permisos sobre los datos

6.7.7. UML (Lenguaje Unificado de Modelado)

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language). Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Objetivos del UML

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

El UML está compuesto por diversos componentes gráficos que se combinan para conformar diagramas cuya finalidad es presentar diversas perspectivas de un sistema a las cuales se les conoce como un modelo.

A continuación se describen los diagramas más comunes del UML y los conceptos que representan.

6.7.7.1. Diagramas de Clases

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones.

Los diagramas de clases permiten a los analistas hablarles a sus clientes en su propia terminología, lo cual hace posible que los clientes indiquen importantes detalles de los problemas que requieren ser resueltos.

Mecanismos de abstracción:

1. Clasificación / Instanciación
2. Composición / Descomposición
3. Agrupación / Individualización
4. Especialización / Generalización

Relaciones entre clases

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

- Asociación y Agregación (vista como un caso particular de asociación)
- Generalización/Especialización.

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Relación de Asociación:

La asociación expresa una conexión bidireccional entre objetos. Una asociación es una abstracción de la relación existente en los enlaces entre los objetos. Puede determinarse por la especificación de multiplicidad (mínima...máxima)

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- * Cero o muchos
- 0..* Cero o muchos
- 1..* Uno o muchos (al menos uno)

Relación de Agregación:

La agregación representa una relación parte de entre objetos. En UML se proporciona una escasa caracterización de la agregación. Esta relación puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes.

Relación de Generalización:

Permite gestionar la complejidad mediante un ordenamiento taxonómico de clases, se obtiene usando los mecanismos de abstracción de Generalización y/o Especialización. La Generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclases heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas. La Generalización y Especialización son equivalentes en cuanto al resultado: la jerarquía y herencia establecidas. Generalización y Especialización no son operaciones reflexivas ni simétricas pero sí transitivas. La especialización es una técnica muy eficaz para la extensión y reutilización.

6.7.7.2. Diagramas de Caso de Uso

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

- Los Casos de Uso (Ivar Jacobson) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el p.d.v. del usuario.
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación.
- Comparación con respecto a los Diagramas de Flujo de Datos del Enfoque Estructurado.
- Los Casos de Uso cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos.
- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo.
- Están basados en el lenguaje natural, es decir, es accesible por los usuarios.

Actores

- Principales: personas que usan el sistema.
- Secundarios: personas que mantienen o administran el sistema.
- Material externo: dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados.
- Otros sistemas: sistemas con los que el sistema interactúa.

Tipos de relación en los Diagramas de Casos de Uso

- Comunicación
- Inclusión: una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino. «include» reemplazó al denominado «uses»
- Extensión: el Caso de Uso origen extiende el comportamiento del Caso de Uso destino. «extend»
- Herencia: el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía.

6.7.7.3. Diagramas de secuencias

Los diagramas de secuencias pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Los diagramas de secuencias se utilizan con frecuencia para representar el proceso descrito en los escenarios de caso de uso. En la práctica, los diagramas de secuencias derivan del análisis de casos de uso y se emplean el diseño de sistemas para generar las interacciones, relaciones y métodos de los objetos del sistema. Los diagramas de secuencias se utilizan para mostrar el patrón general de las actividades o interacciones en un caso de uso. Cada escenario de caso de uso podría crear un diagrama de secuencias, aunque no siempre se crean diagramas de este tipo para los escenarios menores.

6.7.7.4. Diagramas de colaboración

Las colaboraciones describen las interacciones de dos o más cosas en el sistema, las cuales desempeñan en conjunto un comportamiento superior al que puede realizar cualquiera de las cosas por sí sola. Por ejemplo, un automóvil puede dividirse en miles de partes individuales. Las partes se conjuntan para formar los principales subsistemas del vehículo: el motor, la transmisión, el sistema de frenos, etc. Las partes individuales del automóvil se pueden considerar como clases, porque tienen distintos atributos y funciones. Las partes individuales del motor forman una colaboración, porque “colaboran” entre sí para hacer funcionar el motor cuando el conductor pisa el acelerador.

Los diagramas de colaboración muestran la misma información que un diagrama de secuencia, pero su lectura podría ser más difícil. Para denotar las clasificaciones en el tiempo, usted debe indicar un número de secuencia y describir el mensaje.

Un diagrama de colaboración pone énfasis en la organización de los objetos, en tanto que un diagrama de secuencia lo pone en la clasificación de los mensajes según el

tiempo. Un diagrama de colaboración mostrara una ruta para indicar como se enlaza un objeto con otro.

6.7.8 Estudio de Factibilidad

Un estudio de factibilidad trata de determinar la rentabilidad de las distintas alternativas de diseño de información y las prioridades de los diversos componentes del sistema. (Batini & Ceri, 1992)

Los sistemas de factibilidad se apoyan en tres principios básicos: Operativo, Técnico y Económico. Un proyecto debe ser factible en los tres principios para merecer un desarrollo posterior.

Factibilidad significa que el sistema debe auxiliar a la organización a lograr sus objetivos centrales y posibilitar cubrir las metas con los recursos actuales de la organización en los tres principios, tales como: Factibilidad Técnica, Factibilidad Económica y Factibilidad Operativa. (Kendall & Kendall, 2005)

Factibilidad Técnica

- Mejora del sistema actual.
- Disponibilidad de la tecnología que satisfaga las necesidades del usuario.

Factibilidad Económica

- Costo del tiempo del analista de sistemas.
- Costo del estudio.
- Costo del tiempo de los empleados dedicados al estudio.
- Costo estimado del equipo.
- Costo del desarrollo / adquisición del software.

Factibilidad Operativa

- El sistema operará cuando se instale.
- El sistema será utilizado.

6.8 Asociación de Emprendedores Hosanna.

Asociación de Emprendedores de la comunidad de renovación familiar HOSANNA es una asociación Cristo céntrica conformada por emprendedores, empresarios y profesionales enfocados en la restauración y crecimiento de las economías de las empresas y familias, mediante la implantación de los principios establecidos en la palabra de Dios.

Esta asociación como parte del Centro de Capacitación y liderazgo Hosanna (CCLH) tiene como objetivo edificar una generación de hombres, mujeres y jóvenes que no se limitan; sino que van más allá de los recursos que disponen; individuos que se dedican a la búsqueda de oportunidades, desarrollando modelos de negocios bajo los principios del Reino, está orientada a capacitar y equipar a emprendedores, empresarios y profesionales con el fin de proveer las herramientas necesarias para desarrollar y/o potencializar sus talento, proyectos y empresas; de manera que sus vidas sean transformadas, alcanzando así un efecto multiplicador en el nivel y calidad de vida de la comunidad, ciudad, nación y generaciones.

Cuenta un grupo de profesionales altamente calificados nacional e internacionalmente, quienes conforman el claustro de docentes, que se encuentra totalmente comprometidos con la visión, misión y valores de Comunidad Hosanna; asegurando la impartición de principios basados en el Reino de Dios y su establecimiento en la tierra.

Funciona como punto de apoyo para facilitar la innovación y el desarrollo económico. Impulsando el fortalecimiento de las capacidades individuales y colectivas; para alcanzar las aspiraciones de una vida digna, esperanzas de futuro y el justo respeto de los derechos de los Nicaragüenses.

Consta con cuatro programas:

- Programa de desarrollo Humano
- Programa de formación empresarial
- Programa de Idiomas
- Programa de tecnología de la información.

MISIÓN

Establecer el reino de Dios en la vida de cada emprendedor, empresario y profesional con el fin de trabajar, administrar y vivir de acuerdo al diseño originalmente definido por Dios. Edificar una generación de hombres, mujeres y jóvenes que no se limitan; sino que van más allá de los recursos que disponen, desarrollando modelos de negocios bajo los principios del Reino.

VISIÓN

Somos una asociación Cristo céntrica conformada por emprendedores, empresarios y profesionales enfocado en la restauración y crecimiento de las economías de las empresas y familias; mediante la implantación de los principios establecidos en la palabra de Dios. Hombres y mujeres que reproducen el diseño del cielo en la tierra.

VII. DISEÑO METODOLOGICO

7.1 Tipo de Investigación.

El tipo de investigación efectuada fue de tipo Aplicativa, ya que se han implementado un conjunto de conocimientos teóricos y prácticos adquiridos nuestra etapa de formación como profesionales y teniendo como base aquellos cursos de análisis y diseño de sistemas, programación entre otros y el estudio transversal ya que el análisis realizado se efectuó en periodo delimitado relativamente corto. Lo que proporciona todos los elementos necesarios para desarrollar una aplicación móvil que permite brindar soluciones a problemas concretos.

Es a partir de esta investigación que lleva a cabo el desarrollo de una Aplicación Móvil que permite realizar la suscripción de nuevos miembros a la Asociación de emprendedores de la Comunidad Hosanna y el envío de notificaciones de las diferentes actividades de esta escuela de Negocios.

7.2 Población y Muestra.

La población o universo con la que fue realizado el estudio son las escuelas de liderazgo de las iglesias en Managua, tomando como muestra la Asociación de Emprendedores impulsado por comunidad de renovación familiar Hosanna.

7.3 Técnicas e instrumentos de recolección de datos.

Para el desarrollo de la investigación se necesitó obtener información relacionada con el proceso tradicional para la difusión de notificaciones, que funciona a través de envío de correos electrónicos o en la página web de la iglesia o bien a través de boletines informativos de forma impresa generados semanalmente y entregados durante las visitas a las instalaciones de la Iglesia, de esta manera se identifica las problemáticas y necesidades que se buscan resolver.

7.3.1 Técnicas

Durante el proceso de selección y obtención de información se utilizaron las siguientes herramientas:

La encuesta: Admitió realizar un análisis masivo de las personas que quisieran opinar sobre el método actuales y estuvo basadas en la funcionalidad que tendrá la nueva aplicación a desarrollar y así determinar la aceptación por parte de los posibles usuarios

tomando como muestra los alumnos de los cursos impartidos por Asociación de Emprendedores como parte del CCLH (Centro de capacitación y liderazgo Hosanna)

Los resultados obtenidos en la encuesta permitieron conocer el porcentaje de personas que quisieran cambiar el método tradicional y determinar si hay una aceptación de este los nuevos métodos de suscripción y envío de notificación propuesto.

La entrevista: fue realizadas a los responsables del departamento de informática la Asociación de Emprendedores, así como también a aquellas personas involucradas en el tema.

Como resultado de esta entrevista obtuvimos, los requerimientos para el desarrollo de la aplicación; como la información que como organismo quieren presentar en esta aplicación para mantener informados a sus miembros de los cursos y actividades que esta escuela de negocios ofrece.

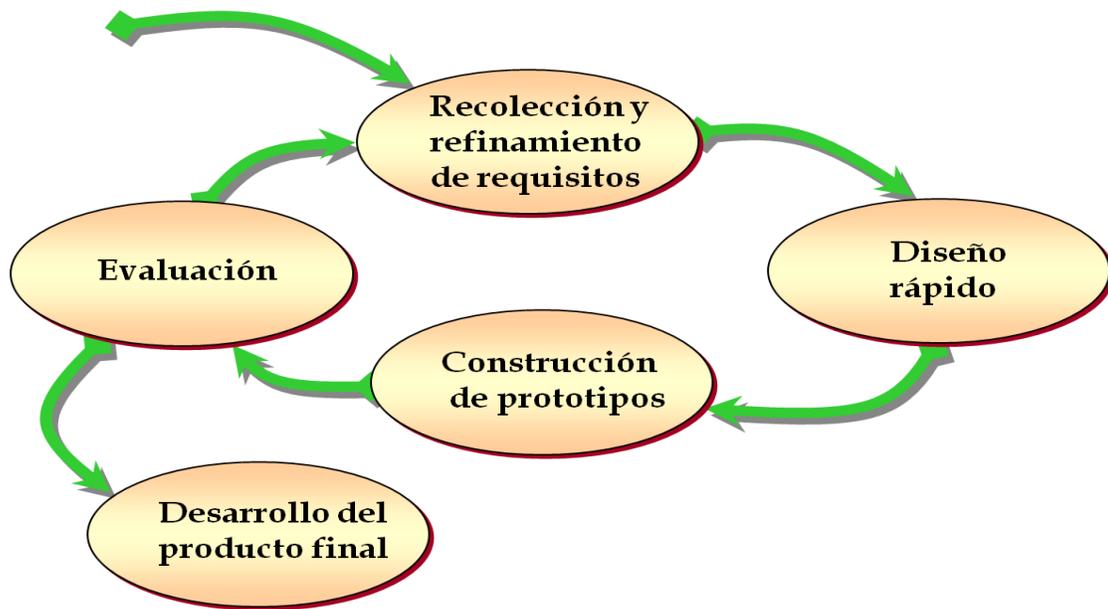
7.3.2 Instrumentos

Para la recolección de la información de la presente investigación los instrumentos que se utilizaron consiste en un cuestionario que contiene preguntas cerradas y de opción múltiple en el caso de la encuesta y en la entrevista un cuestionario de preguntas cerradas y abiertas, para la estructuración de estos instrumentos se tomó en cuenta aspectos como: Sistemas operativos a los que los usuarios potenciales cuentan en sus móviles, aceptación de la tecnología y objetivos de desarrollo de la aplicación.

7.4 Modelo de desarrollo.

Luego de la entrevista con los encargados de la asociación de emprendedores, se definió los objetivos globales de la aplicación a desarrollar, y decidió que el modelo de desarrollo que más se ajustaba a las necesidades del proyecto a realizar era el prototipo evolutivo, debido a las características de este; que permite ver las reacciones y sugerencias del cliente durante el proceso de elaboración y realizar las modificaciones en una etapa temprana del desarrollo y con el menor trastorno posible, envase a la retroalimentación que conlleva al producto final.

En el desarrollo de la aplicación se cumplió con cada una de las etapas de modelo de desarrollo seleccionado llevándose a cabo dos ciclos de iteración:



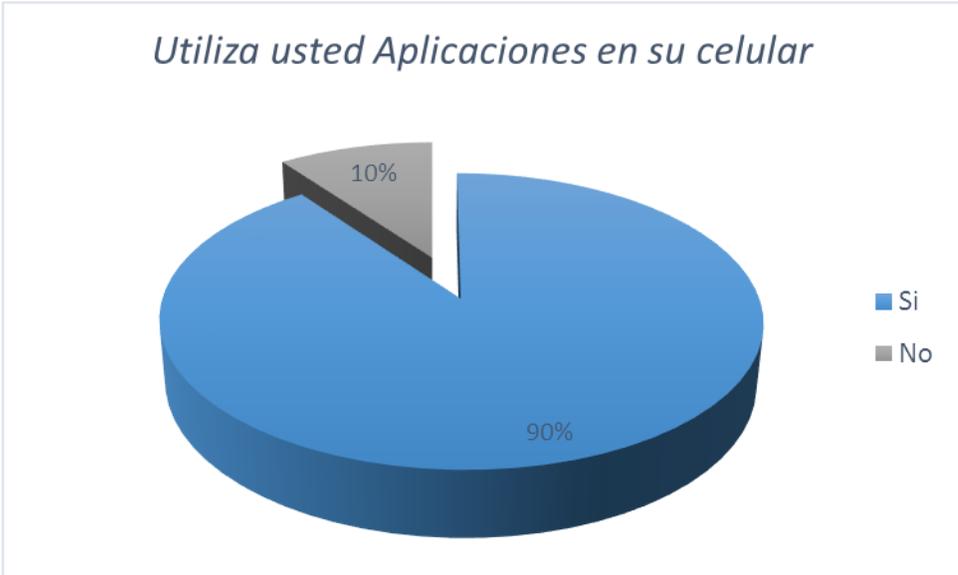
En el primera iteración en la etapa de comunicación con los encargados del área de informática de Asociación de Emprendedores Hosanna (el cliente) se obtuvieron los requerimientos necesario para desarrollo del presente proyecto, lo cual conlleva a la segunda etapa que se trató del diseño rápido de las actividades (Pantallas) con las que el usuario final interactuaría y luego la construcción del primer prototipo, para la respectiva evaluación, en la segunda iteración en base a la retro alimentación por parte del cliente se rediseño la parte de la suscripción permitiéndoles a los usuarios al seleccionar el tipo de información desea visualizar en su móvil, por medio de un menú que muestra los servicios o actividades que el organismo ofrece y en la pantalla principal dar la opción de informarse acerca de que es Asociación de Emprendedores, la misión y la visión del mismo.

VIII. ANALISIS DE RESULTADO

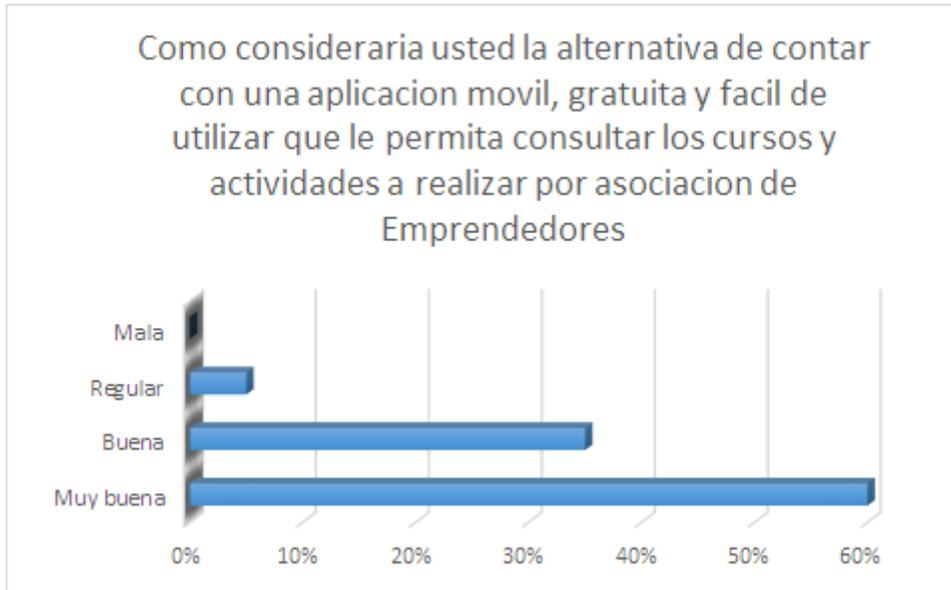
8.1 Encuesta



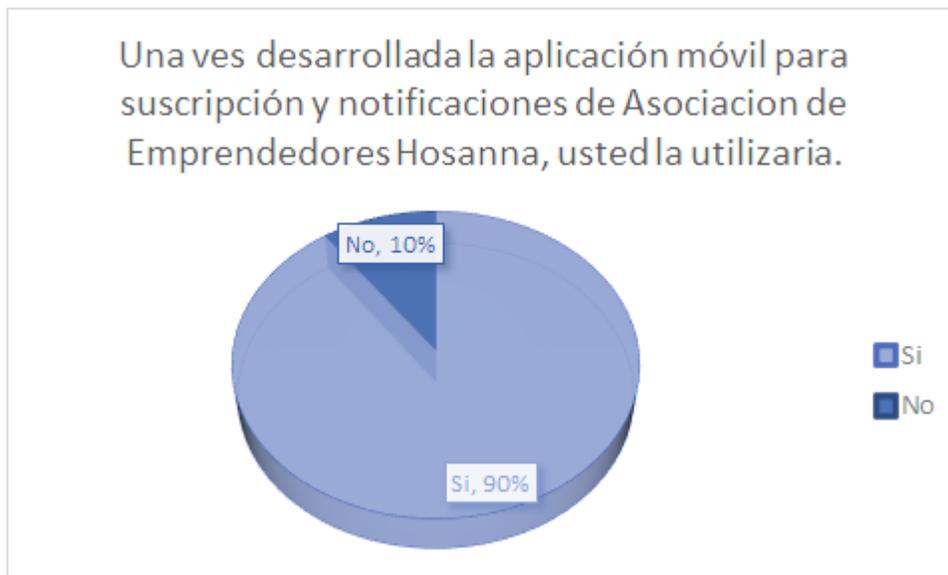
El resultado de la encuesta delevó que la mayoría de las personas en la actualidad optan por un Smartphone con OS Android, debido a que es un software libre, tienes más opciones para descargar aplicaciones gratuitas.



El noventa por ciento (90%) de las personas encuestadas respondieron que si hacen uso de aplicaciones en su celular.



Un sesenta por ciento (60%) de las personas encuestadas calificaron como una alternativa muy buena contar con una aplicación móvil, gratuita y fácil de utilizar, que permita consultar los cursos y diferentes actividades que Asociación de Emprendedores realiza y un treinta por ciento (30%) calificó esta alternativa como buena lo que sumado nos da un total de noventa por ciento (90%) de aceptación de la nueva aplicación.



Un noventa por ciento (90%) de las personas en encuestadas mostraron disponibilidad a utilizar la aplicación móvil para suscripción y notificaciones de Asociación de emprendedores.

8.2 IDE (Entorno de Desarrollo Integrado)

Definido el Sistema operativo para el desarrollo de la aplicación móvil fue necesario evaluar los entornos de desarrollo a utilizar para la aplicación móvil que dio como resultado el siguiente cuadro comparativo de los IDE para Android.

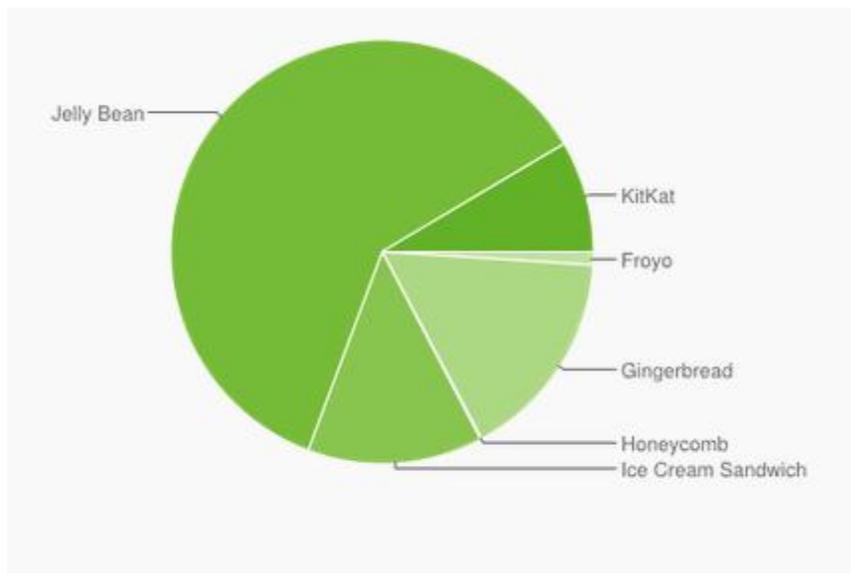
Eclipse IDE	IntelliJ IDEA (Android Studio)
Más usado	Más ligero
Mayor estabilidad	Es una versión Beta
Gratuito	De pago para uso comercial
Es fácil encontrar soluciones al llevar más tiempo funcionando	Se dificulta encontrar ayuda con un error o el propio IDE da mensajes contradictorios
Mayor velocidad	Compilación más lenta.
Compilación limpia: 8 seg	Compilación limpia: 12 seg
Compilación incremental: 3 Seg	Compilación incremental 5 seg
Autocompletado rígido	Autocompletado contextual
Cuenta con Herramientas necesarias para desarrollar componentes NDK en C o C++	No soporta desarrollo para NDK,

Aunque Android Studio se platea como el entorno de desarrollo del futuro y todo indique que será lo que los desarrolladores recomienden; en la actualidad está en versión beta y tiene muchos errores, en la actualidad hay más información de soluciones en el IDE Eclipse para Android que en el mismo Android Studio, es por esta razón que se decidió que la mejor opción para desarrollar la presente investigación es eclipse.

8.3 Distribución de las versiones del OS Android.

Según la página oficial de desarrollo Android (<https://developer.android.com/about/dashboards/index.html>) en mayo del presente año, la distribución de las diferentes versiones de Android se encuentra de la siguiente manera:

Versión	Nombre Comercial	API	Distribución
2.2	Froyo	8	1.0%
2.3.3 - 2.3.7	Gingerbread	10	16.2%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	13.4%
4.1.x	Jelly Bean	16	33.5%
4.2.x		17	18.8%
4.3		18	8.5%
4.4	KitKat	19	8.5%



Se decidió realizar el presente proyecto bajo la versión 4.0.3 ya que según la investigación previa esta versión es compatible con 82.7% de las demás versiones según su distribución, además de la característica con la que cuenta de unificar las dos versiones

anteriores (2.x para teléfonos y 3.x para Tablets) en una sola compatible con cualquier tipo de dispositivo.

8.4 Servicio Web.

Se realizó un servicio web; para permitir el acceso remoto a datos de la aplicación móvil al servidor, en el análisis realizado se encontró con varias tecnologías que permitían, realizar esta misma tarea con el entorno de sistemas distribuidos funcionando sobre internet, pero estas tecnologías presentan limitantes debido a la heterogeneidad del entorno, la presencia de firewalls y al uso de formatos propietarios para representar datos.

La siguiente tabla muestra un resumen de las diferentes tecnologías de procesamiento distribuido.

Sistemas de Comunicación	DCOM	CORBA	Java RMI	Web Services
Protocolo	RPC	IIOP	IIOP ó JRMP	HTTP
Formato de mensajes	NDR	CDR	Java ser. Format	SOAP
Descripción	IDL	OMG IDL	Java	WSDL
Descubrimiento	Windows Registry	Naming Service	RMI Registry ó JNDI	UDDI

Descubrimiento: es el proceso de localizar uno o varios documentos que describen un determinado servicio Web XML mediante el Lenguaje de descripción de servicios Web

Según la evaluación realizada se determino que bajo estas condiciones los servicios web y su protocolo de comunicación SOAP es una alternativa sumamente útil para comunicar aplicaciones heterogéneas (interoperabilidad), fundamentalmente por el uso de XML que es un estándar basado en texto e independiente de plataformas y lenguajes y por el uso de HTTP como transporte, el cual normalmente atraviesa los firewalls dado que estos no bloquean el puerto HTTP. Además al no ser un protocolo sofisticado y al no definir modelos de programación permite que las aplicaciones tengan un bajo acoplamiento, lo cual es ideal en el entorno de internet.

8.5 Herramientas Utilizadas.



IDE-Eclipse: Entorno de desarrollo utilizado, para realizar la aplicación.



PhotoShopen online: se utilizó para el diseño y retoque de imágenes.



ER/Studio: Herramienta utilizada en el modelado de datos



Visual estudio 2010: utilizado para el desarrollo del servicio web que permitirle a la aplicación móvil comunicarse con la base de datos y el diseño de los diagramas UML que describen la funcionalidad de la aplicación.



Lenguaje java: utilizado para la programación de la aplicación móvil.



SQL Server 2008: utilizado en desarrollo de la base de datos que almacena la información a visualizar en la aplicación móvil.



Office 2007: utilizado para la elaboración del documento del presente proyecto.

8.6 Descripción de la aplicación.

La Aplicación móvil para la suscripción y notificaciones de la asociación de emprendedores de la comunidad de renovación familiar Hosanna, 2014; fue desarrollada bajo la plataforma Android 4.0.3 Ice Cream Sandwich, es compatible con las versiones posteriores, esta aplicación se comunica con la base de datos por medio de un servicio web desarrollado en visual studio 2010, la que permite realizar al usuario la suscripción de manera más eficaz a esta asociación y consultar las diferentes actividades o cursos que esta escuela brinda.

Beneficios:

- ✓ Mejora el canal de comunicación de la asociación con cada uno de sus miembros permitiendo a sus miembros consultar los cursos en cualquier lugar y en cualquier momento.
- ✓ La realización de la suscripción más eficaz y permite a la asociación tener un registro actualizado de la cantidad de personas que están siguiendo las actividades que realiza, por medio de la aplicación.
- ✓ Permite promocionar los cursos que ofrece la asociación.
- ✓ Es un nuevo punto de contacto con los miembros de la asociación.

8.7 Estudio de Factibilidad

Una vez definida la problemática del estudio, se establecieron las causas que conllevaron al desarrollo de la aplicación móvil en la plataforma Android, la cual fue sometida a un estudio de factibilidad que determina la viabilidad del proyecto. Es decir, se determinaron los requerimientos tecnológicos, económicos y operacionales para el desarrollo de dicho proyecto, Este estudio ayuda a establecer los beneficios y el grado de aceptación que tendrá la propuesta ante los usuarios finales.

El estudio de factibilidad se divide en tres partes principales: Factibilidad Técnica, Económica y Operacional.

8.7.1 Factibilidad Técnica

La factibilidad técnica nos permitirá evaluar si el hardware y software están disponibles para que pueda satisfacer las necesidades requeridas para el desarrollo de la aplicación.

En este estudio se especifican los equipos técnicos o equipos informáticos de hardware y software que pueda satisfacer las necesidades requeridas para el desarrollo e implementación de la aplicación móvil.

- Equipos de cómputo y dispositivos necesarios para el desarrollo de la aplicación móvil.

Equipo	Características	Cantidad
Servidor WEB	✓ Procesador: Intel (R) Celeron(R) CPU 430 @ 2.60 GHz 2.60 GHz ✓ Memoria RAM: 8GB ✓ Capacidad de Disco duro 500GB	1
Computadora	✓ procesador: Intel(R) Celeron(R) CPU 430	1

de Escritorio	@1.80 GHz 1.79GHz ✓ Memoria RAM de 2 GB ✓ Capacidad Disco Duro 80 GB	
Laptop	✓ procesador: Intel(R) Cor(TM) CPU 430 @1.80 GHz 1.79GHz ✓ Memoria RAM de 4 GB ✓ Capacidad Disco Duro 500 GB	1
Smartphone LG Optimus L3 II	✓ Tipo LCD IPS touchscreen capacitivo, 16M colores ✓ Tamaño 240 x 320 pixeles, 3.2 pulgadas ✓ Soporte multitouch ✓ Velocidad de datos 32 - 48 kbps ✓ OS: Android OS, v4.1.2 Jelly Bean ✓ Procesador 1GHz GPU Adreno 2000 ✓ VERSION DEL KERNEL3.4.0	1

- Requerimientos de Software para el desarrollo de aplicación móvil

Paquete	Descripción
Microsoft Windows 7 Profesional	Sistema Operativo para computadoras
Microsoft Office 2007	Paquete de Oficina(suite ofimática)
Microsoft Visual Studio 2010	Software para Programación de Sistemas, esta Aplicación nos ayuda hacer diagramas UML y otros más, todos ellos enfocados en el área del software. Los diagramas hechos con esta aplicación son:

	<input type="checkbox"/> Diagrama de casos de uso <input type="checkbox"/> Diagrama de clases <input type="checkbox"/> Diagrama de secuencia
SQL Server 2008	Es un sistema de gestión de bases de datos relacional que integra Desarrollo de Software, Administración de bases de Datos, Diseño de Bases de Datos, Creación y Mantenimiento.
ERStudio	Software para Diseño de Diagramas
JDK (Java Development Kit).	Herramientas de desarrollo para la creación de programas en Java.
Eclipse IDE for Java Developers	Entorno de desarrollo por excelencia en Android y Java
SDK de Android.	Android SDK es el kit de desarrollo de Google para su sistema operativo Android, y gracias a él es muy sencillo virtualizar un sistema operativo Android en tu PC.
El plugin ADT	Permite crear una AVD (Android Virtual Device) Dispositivo Virtual Android para lanzar y depurar la aplicación.

8.7.2 Factibilidad Económica

Este estudio nos permite proporcionar la medida de los costos en que se incurren en la realización del proyecto a desarrollar.

- Estimación de costos de los equipos de Técnicos para el desarrollo y prueba de la aplicación móvil.

Equipo	Cantidad	Precio Unitario	Sub-Total
Servidor WEB	1	\$600,00	\$600,00
Computadora de Escritorio	1	\$300,00	\$300,00
Laptop HP 450	1	\$500,00	\$500,00
Smartphone LG 2g optimus	1	\$150,00	\$150,00
		Total	\$1550,00

- Estimación de costos del software para el desarrollo Aplicación móvil.

Paquete	Cantidad	Precio Unitario	Sub-Total
Microsoft Windows 7 Profesional	1	\$300,00	\$300,00
Microsoft Office 2007	1	\$229,00	\$229,00
Microsoft Visual Studio 2010	2	\$399,00	\$798,00
SQL Server 2008	2	\$200,00	\$400,00
ERStudio	1	\$0,00	\$0,00
JDK (Java Development Kit).	2	\$0,00	\$0,00
Eclipse IDE for Java Developers	2	\$0,00	\$0,00
Total			\$1,727,00

- Estimación de costos del personal y otros insumos de desarrollo de aplicación móvil.

Descripción	Sub-Total
Costos Operativos	
Costo de mano de obra	\$650,00 (Costo Estimado)
Materiales y suministros	
Papelería y otros insumos	\$100,00 (Costo Estimado)
Total	\$750,00

8.7.3 Factibilidad Operacional

Para este estudio se tomaron en consideración los resultados obtenidos de las herramientas de obtención de datos, tales como la entrevista, las cuales fueron aplicadas a miembros que asisten a la asociación de emprendedores de la iglesia Hosanna que son usuarios potenciales de la nueva aplicación, de manera que se identificaron las necesidades y se da respuesta a las problemáticas definidas.

A demás se ha mantenido una comunicación constante con los encargados del área informática y personas interesadas para probar los avances del aplicativo y se han tomado en cuenta los cambios o sugerencias indicados por estos.

Con el desarrollo de la aplicación móvil se realizaran las funciones que el usuario le solicite manera fácil y rápida, a través del diseño adecuado.

8.8 Diagramas UML .

8.8.1. Diagramas de Caso de USO

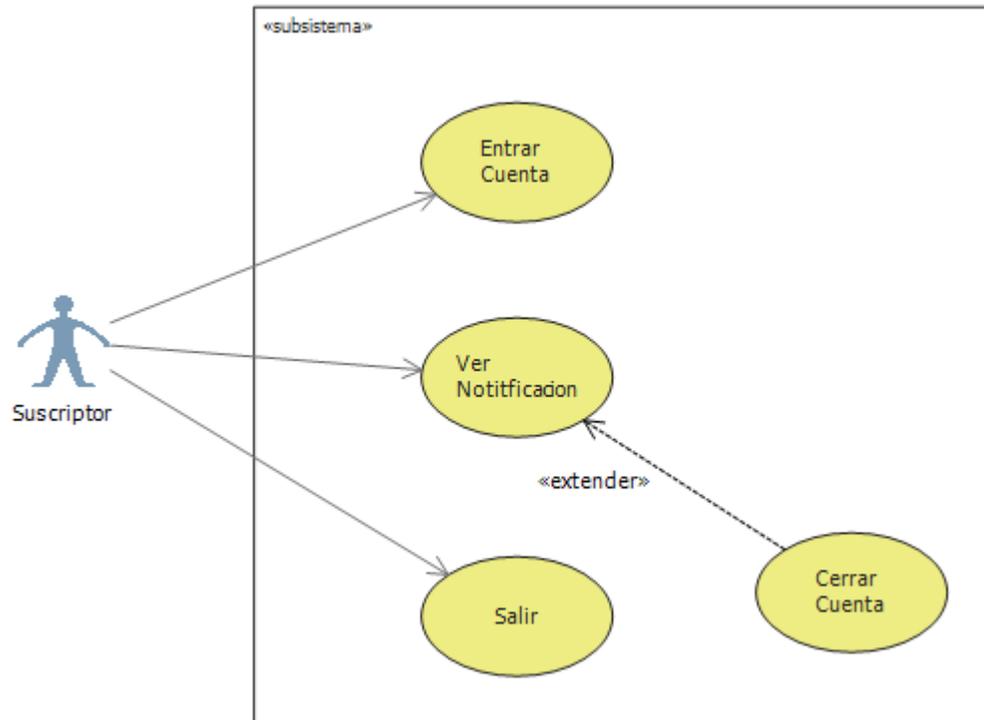
El usuario Realiza la Suscripción

▣

El Administrador Realiza La Notificación

☒

El Suscriptor Entra a su cuenta para ver notificaciones



8.8.2. Diagramas de Secuencia

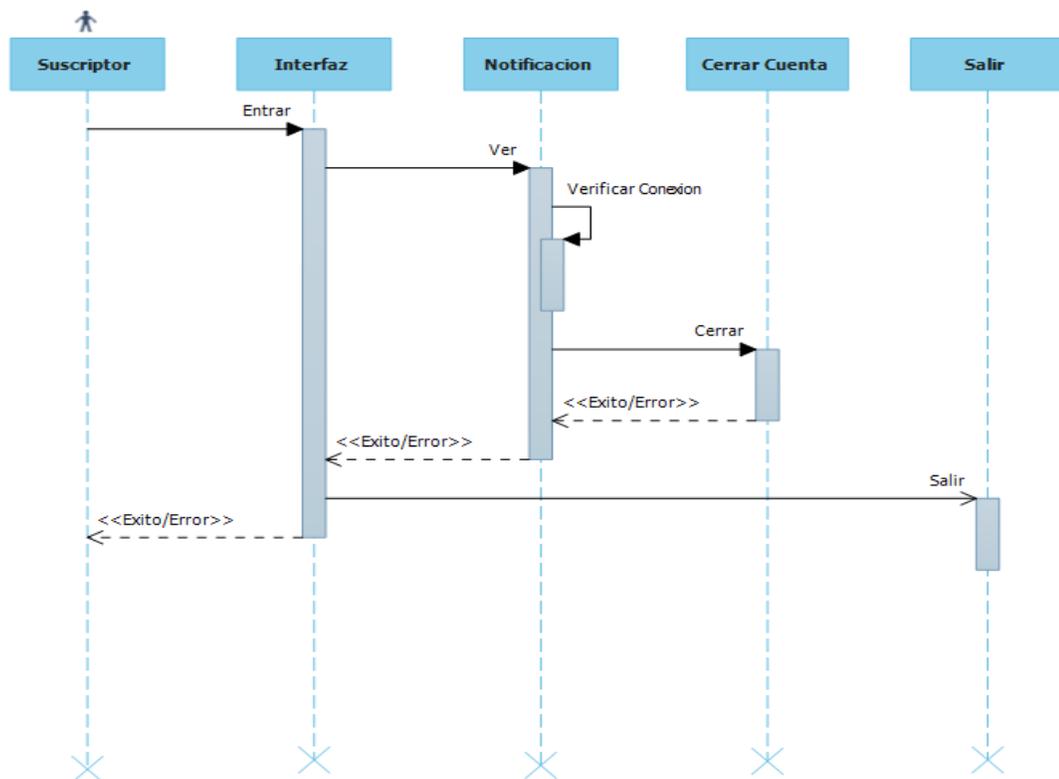
- El usuario realiza suscripción.

▣

- El Usuario Visualiza las Notificaciones

PH

- El Usuario cierra cuenta



- El Administrador Crea Notificación.
x

- El administrador Modifica Notificación.

x

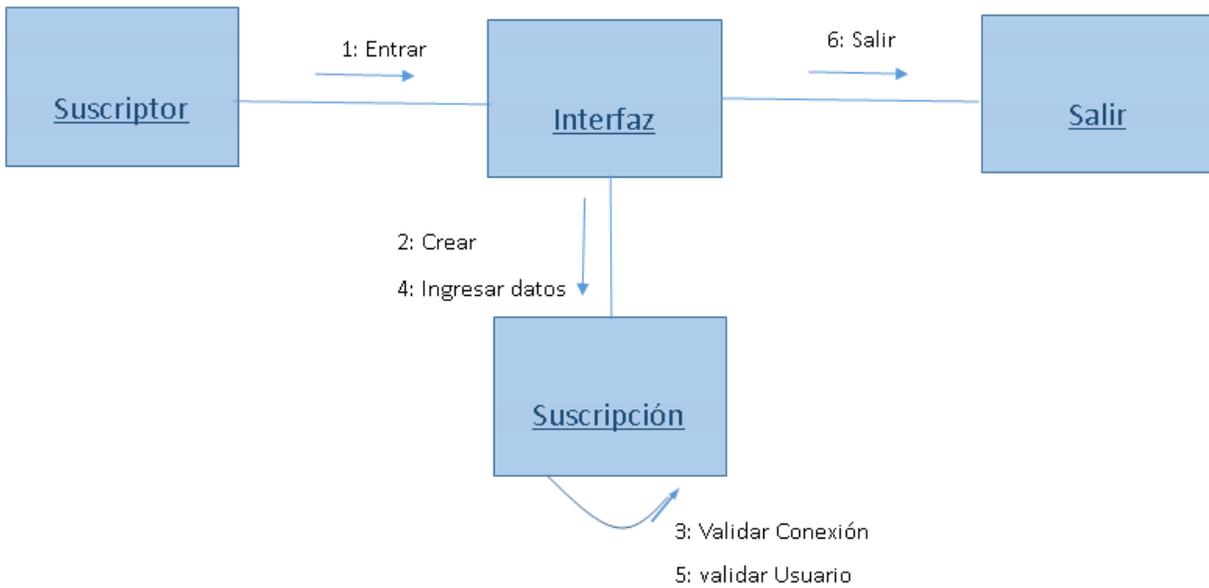
- El Administrador elimina Notificación.

8.8.3. Diagrama de Clases

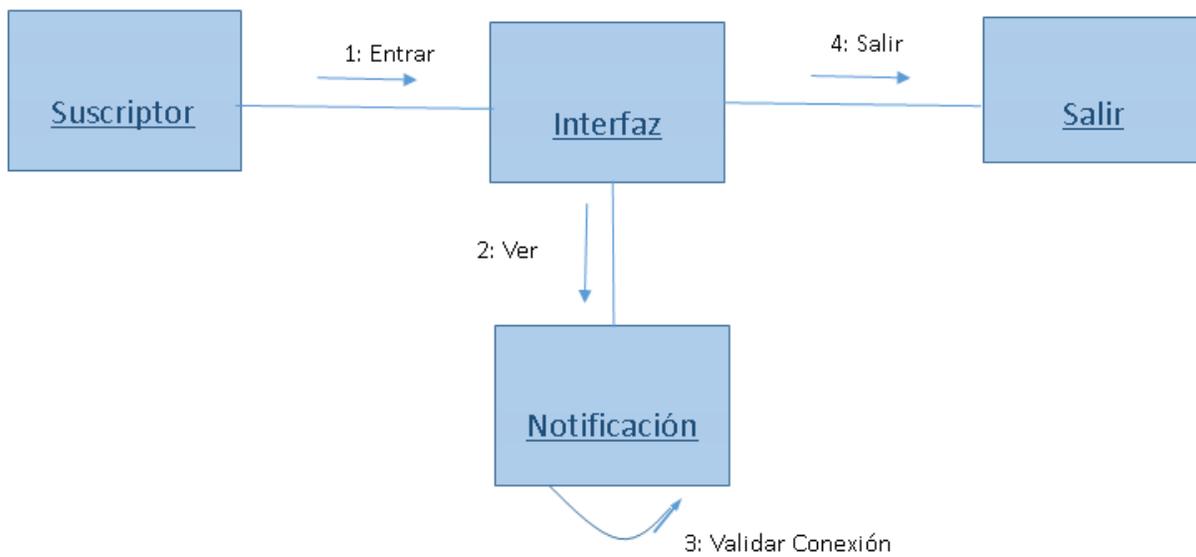
14

8.8.4 Diagrama de Colaboración

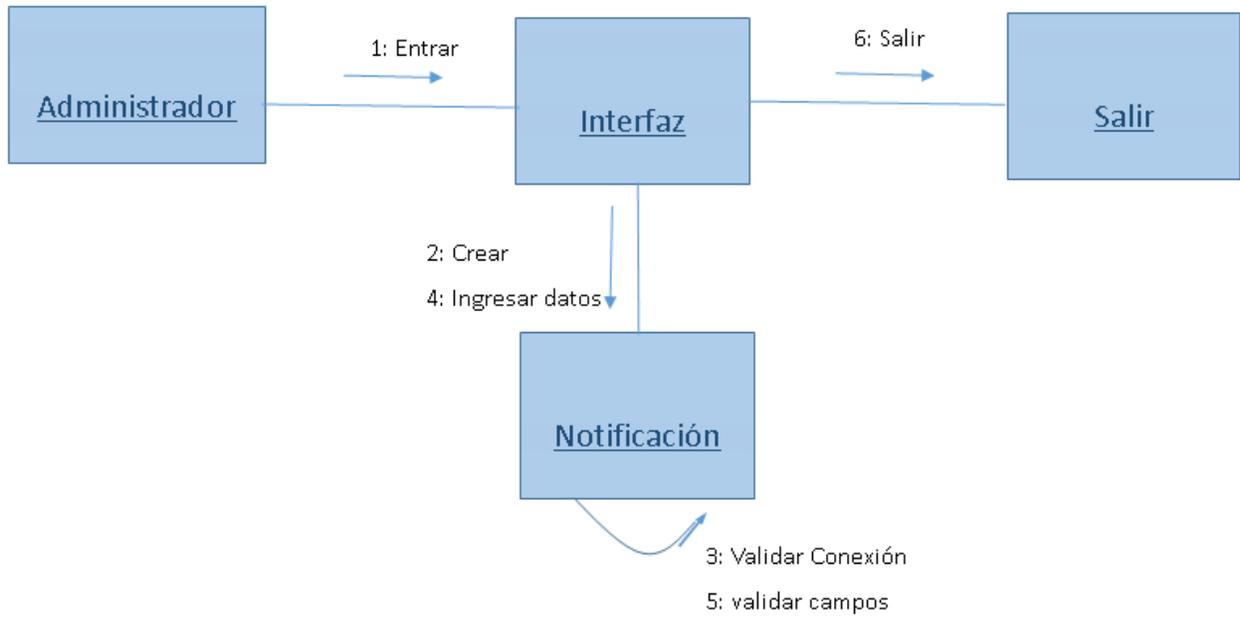
- El usuario realiza Suscripción



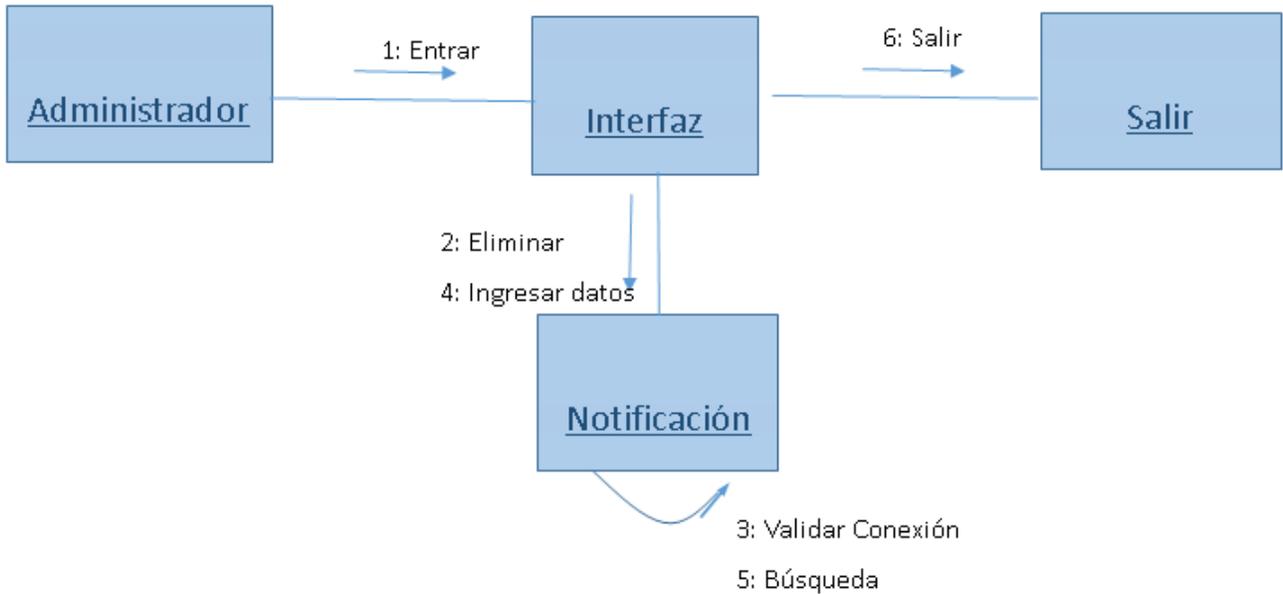
- El usuario Visualiza las Notificaciones



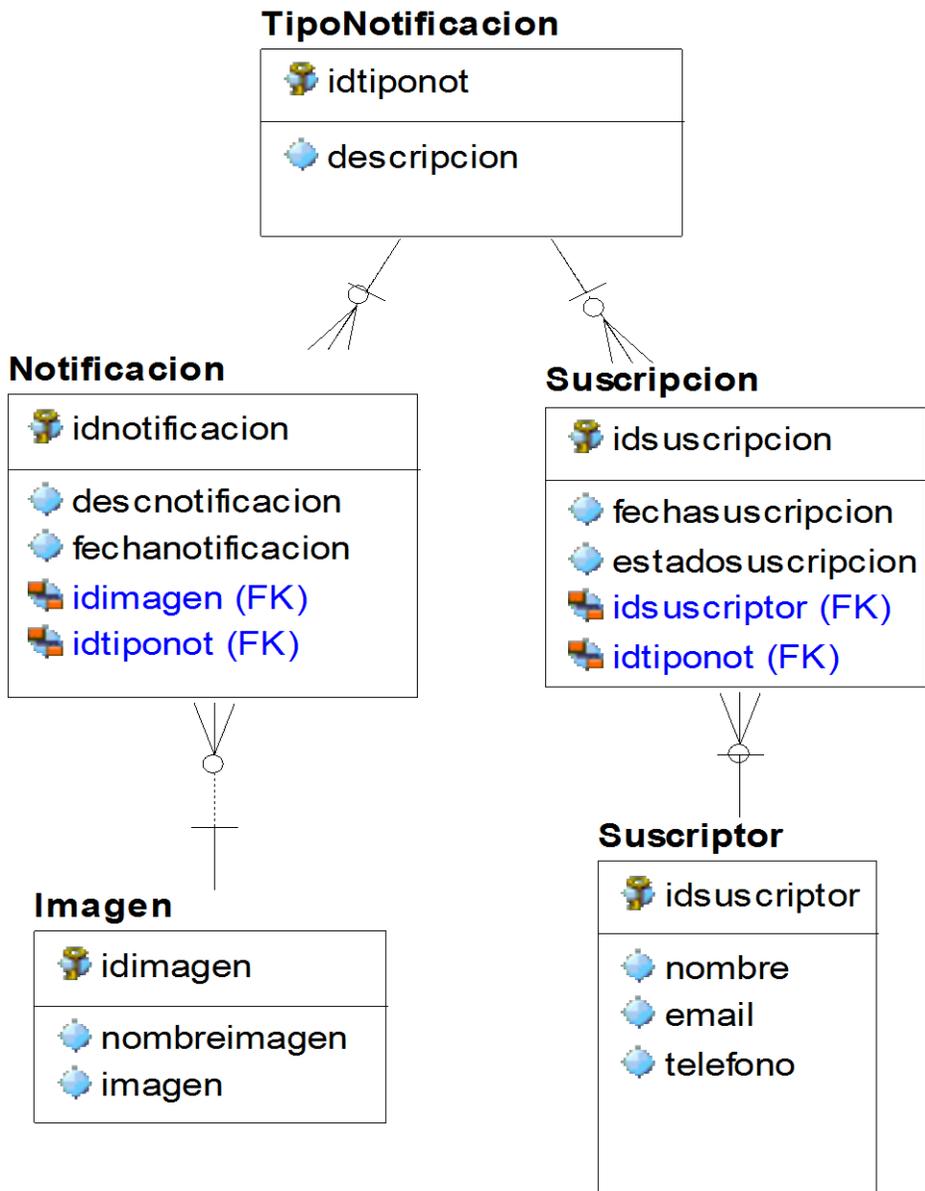
➤ El administrador crea la Notificación



➤ El administrador Elimina notificación



8.8.5 Diagrama E/R



8.9 Diccionario de Datos

Identidad:



Notificación

Nombre del Atributo	Definición	Tipo de Dato
idnotificacion	Número de identificación de la notificación	INTEGER
descnotificacion	Descripción o características de la notificación, el campo puede ser nulo.	NVARCHAR(50)
fechanotificacion	Fecha de creación de la notificación	DATETIME
idimagen	Numero identificador de la imagen	INTEGER
idtiponot	Número que identifica una notificación según su tipo	INTEGER



Tiponotificacion

Nombre del Atributo	Definición	Tipo de Dato
idtiponot	Número que identifica una notificación según su tipo	INTEGER
descripción	Nombre o clasificación de las notificaciones	NVARCHAR(50)



Suscriptor

Nombre del Atributo	Definición	Tipo de Dato
idsuscriptor	Número de identificación del suscriptor	INTEGER
nombre	Nombre Completo del suscriptor	NVARCHAR(50)
email	Correo electrónico del suscriptor	NVARCHAR(50)
teléfono	Número de teléfono del suscriptor	NVARCHAR(8)



Suscripción

Nombre del Atributo	Definición	Tipo de Dato
idsuscripcion	Numero de suscripción	INTEGER
fechasuscripcion	Fecha de creación de la suscripción	DATETIME
estadosuscripcion	Estado Activo o inactivo de la suscripción	INTEGER
idsuscriptor	Número de identificación del suscriptor	INTEGER
idtiponot	Número que identifica una notificación según su tipo	INTEGER



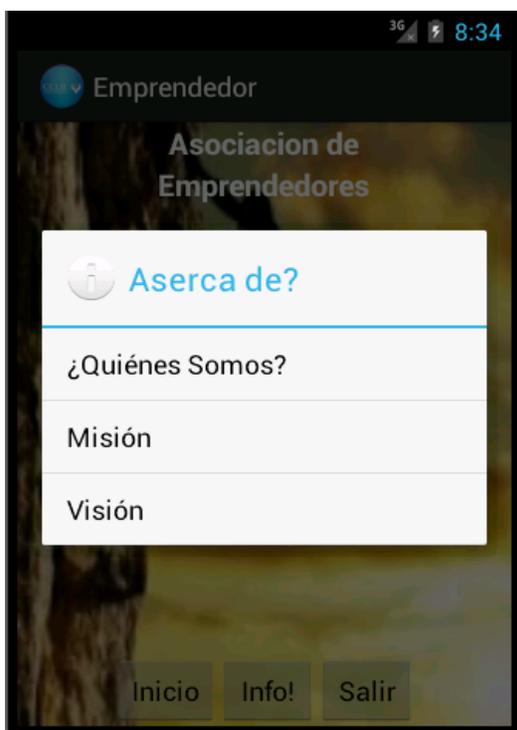
Imagen

Nombre del Atributo	Definición	Tipo de Dato
idimagen	Identificador de la imagen, auto incremental	INTEGER
nombreimagen	nombre de la imagen	VARCHAR(50)
imagen	Almacena la ruta de la imagen	NVARCHAR(max)

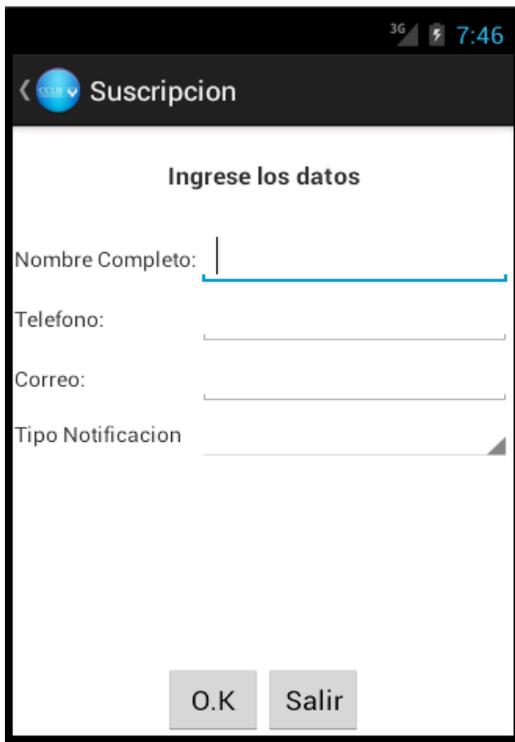
8.10 Pantallas de la aplicación.



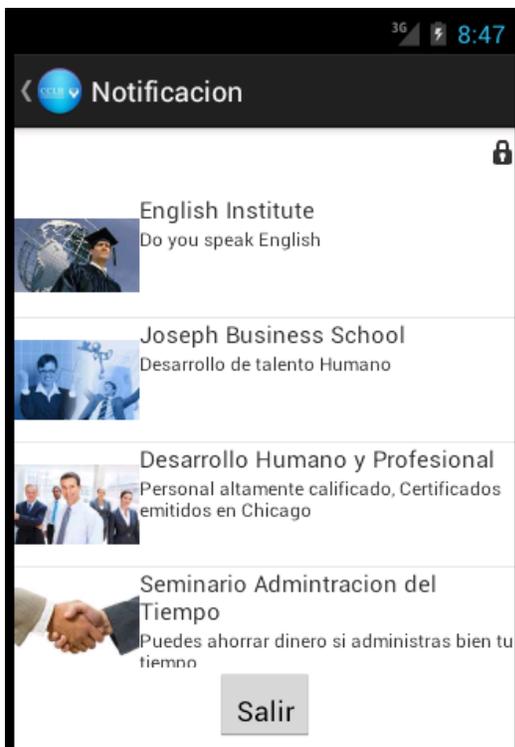
Pantalla de presentación de la aplicación, presenta tres opciones Inicio; ingresa al contenido de la aplicación, info!; presenta un menú que permite ver información relacionada con el organismo y salir.



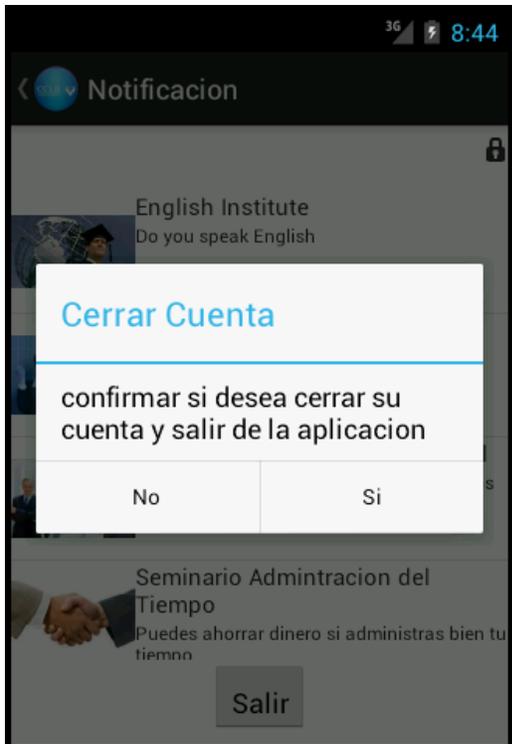
Menú acerca de: presentado al pulsar el botón info! de la pantalla de presentación de la aplicación, en la que el usuario puede informarse a cerca de que se trata el organismo su misión y visión



Pantalla de suscripción, presenta el formulario, con la información que se solicita para la suscripción para recibir las notificaciones según el tipo seleccionado, la opción OK que permite almacenar la información en la base de datos y la opción salir en casos que desista de brindar los datos.



Pantalla de notificaciones presenta la lista de notificaciones disponibles según la opción seleccionada por el usuario, que consta de una imagen asociada a la notificación un título y un subtítulo o breve descripción, un botón con la opción de salir de la aplicación y un ImageView que brinda la opción de cerrar la suscripción.



Notificación que solicita la confirmación para cerrar la suscripción al confirmar, se deshabilita la suscripción y se cierra la aplicación.

8.11 Pruebas realizadas.

Se realizaron pruebas con la aplicación en los siguientes dispositivos:

- ✓ Smartphone: LG Optimus L3 II Android OS, v4.1.2 Jelly Bean, pantalla de 3.2 pulgadas.
- ✓ Smartphone: LG Optimus L7 Android OS, v4.0 Ice Cream Sandwich, pantalla de 4.3 pulgadas.
- ✓ Smartphone: Alcatel 4015A, v4.2.2 Jelly Bean, pantalla 3.5 pulgadas.
- ✓ Tablet: Titán, Pc 7052ME Android OS, v4.2.2 Jelly Bean, pantalla 7 pulgadas 1024x600.

IX. CONCLUSIONES

Se desarrolló la Aplicación móvil para la suscripción y notificaciones de la Asociación de Emprendedores de la comunidad de renovación familiar “HOSANNA”, año 2014, bajo la plataforma Android; Iniciando con una investigación sobre el Sistema Operativo Android, su arquitectura, Herramientas y entorno de desarrollo, Para ello luego de la investigación previa se evaluó el entornos de desarrollo disponibles para el desarrollo de aplicaciones móviles bajo la plataforma Android la cual dio como resultado la elección del entorno IDE Eclipse por ser más estable y encontrarse mayor información de soluciones de los errores que este puede presentar.

Una vez definido el entorno de desarrollo, los requerimientos necesarios para la elaboración de la aplicación se realizó el diseño y programación de la aplicación utilizando la plataforma Android 4.0.3 Ice Cream Sandwich y el lenguaje de programación Java.

Para valorar la aceptación y viabilidad del proyecto, se realizó:

- Una entrevista con responsables del departamento de informática la Asociación de Emprendedores, y con algunas personas involucradas en el tema en la cual se pudo definir los objetivos generales de la aplicación.
- Una Encuesta a usuarios potenciales de la aplicación en la cual un 70 por ciento de las personas encuestadas demostraron aceptación e interés en la implementación de dicha aplicación.

X. RECOMENDACIONES

- ✓ Agregar más funcionalidades a la aplicación para lograr una mejor interacción con los usuarios.
- ✓ Desarrollar la aplicación en diferentes plataformas y a si alcanzar los usuarios que cuentan con otro tipo de sistema operativo móvil.
- ✓ Garantizar la distribución de la aplicación en la página web de la asocian de emprendedores.

XI. BIBLIOGRAFIA

LIBROS

- Batini, C., & Ceri, S. (1992). *Diseño Conceptual de base de Datos*. Addison Wesley.
- Kendall, k., & Kendall, J. (2005). *Analisis y Diseño de Sistemas* (6ta ed.). Mexico: Pearson Prentice Hall.
- Pressman, R. (2005). *Ingenieria del Software Un enfoque Practico* (6ta ed.). Mc Graw Hill.

WEBGRAFIA

- *Aplicaciones Moviles, La Evolucion*. (14 de 09 de 2013). Recuperado el 06 de 05 de 2014, de <http://upsasoyyo.wordpress.com/>:
<http://upsasoyyo.wordpress.com/2013/09/17/aplicaciones-moviles-la-evolucion/>
- Aranaz Tudela, J., & Campo Vazques, C. (2009). *bitstream*. Recuperado el 13 de 04 de 2014, de Desarrollo de AplicacionesPara dispositivos moviles sobre la platoforma Android de Google : [http://e-archivo.uc3m.es/bitstream/handle/10016/6506/PFC Jaime_Aranaz_Tudela_2010116132629.pdf?sequence=1](http://e-archivo.uc3m.es/bitstream/handle/10016/6506/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf?sequence=1)
- Baz Alonso, A., Ferreira Artime, I., Alvarez Rodrigues, M., & Garcia Baniello, R. (2009). *Dispositivos Moviles*. Recuperado el 15 de 4 de 2014, de Dispositivos moviles: <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>
- *Definicion.de*. (2010). Recuperado el 19 de 04 de 2014, de Definicion de Programacion: <http://definicion.de/programacion/#ixzz2zA8xsVe0>
- Frida, P. (24 de 03 de 2009). *Apuntes de informatica*. Recuperado el 18 de 04 de 2014, de Algoritmo: <http://informaticafrida.blogspot.com/2009/03/algoritmo.html>
- Gonzáles, F. (05 de 12 de 2013). *github.com*. Recuperado el 26 de 05 de 2014, de Conceptos Basicos de SQL: https://github.com/geotalleres/geotalleres/blob/master/conceptos-sql/conceptos_sql.rst
- GP, P. (19 de 03 de 2014). *blogspot.com*. Recuperado el 20 de 04 de 2014, de Desarrolla Aplicaciones moviles: <http://aplicacionesmoviles-402.blogspot.com/2014/03/tipos-de-sistemas-operativos.html>

- *Ingenieria del Software*. (12 de 8 de 2009). Obtenido de [blogspot.com: http://ingsoftware2udistrital.blogspot.com/2009/08/modelo-de-proceso-prototipos.html](http://ingsoftware2udistrital.blogspot.com/2009/08/modelo-de-proceso-prototipos.html)
- Martinez, I. (14 de 03 de 2014). *blogspot.com*. Recuperado el 20 de 04 de 2014, de Aplicaciones moviles: http://dispositivosmovilesa.blogspot.com/2014_03_01_archive.html
- Mcornielly. (25 de 09 de 2009). *Scribd*. Recuperado el 2014 de 04 de 18, de Lenguajes de Programacion: <http://es.scribd.com/doc/20189291/LENGUAJE-DE-PROGRAMACION>
- *Tipos de investigacion*. (2008). Obtenido de [Rena.edu.ve](http://www.rena.edu.ve/): <http://www.rena.edu.ve/cuartaEtapa/metodologia/Tema4.html>
- Nuñez, K. (18 de 07 de 2013). <http://es.slideshare.net/>. Obtenido de Sistema Operativo Android: Versiones. Historia: <http://es.slideshare.net/karenonunez/sistema-operativo-android-versiones-historia>
- Parra, M. (05 de 2013). sistemas de software. Recuperado el 02 de 04 de 2014, de [blogspot.com: http://sistemasdesoftware2actividad.blogspot.com/2014/05/modelos-evolutivos-de-software.html](http://sistemasdesoftware2actividad.blogspot.com/2014/05/modelos-evolutivos-de-software.html)
- Roldan, M. (20 de 09 de 2013). *mibqyyo*. Recuperado el 19 de 04 de 2014, de desarrollo de aplicaciones en android II- Herramienta sdk: http://www.mibqyyo.com/articulos/2013/09/20/desarrollo-de-aplicaciones-en-android-ii-herramientas-sdk/#/vanilla/discussion/embed/?vanilla_discussion_id=0
- Saffirio, M. (05 de 02 de 2006). *wordpress.com*. Obtenido de Que son los Web Services: <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>
- Salvador Gomez, O. (11 de 2011). *www.sgoliver.net*. Recuperado el 17 de 05 de 2014, de Curso de Programacion Android: http://www.sgoliver.net/blog/?page_id=2935
- Universidad Carlos III-MAdrid. (s.f.). *sites.google.com*. Recuperado el 18 de 04 de 2014, de Arquitectura de android: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

XII ANEXOS

FORMATO DE ENTREVISTA

Entrevista a los representantes de Asociación de Emprendedores Hosanna.

La presente entrevista fue realizada con el objetivo de definir los objetivos globales de la aplicación para la asociación de emprendedores. Como trabajo de investigación para la tesis de seminario de graduación del departamento de computación.

- ¿Cuál es la misión/Visión de Asociación de Emprendedores?
- ¿Hay algún requisito para pertenecer a la asociación de Emprendedores?
- ¿La Asociación de emprendedores es exclusivo para miembros de comunidad hosanna?
- ¿Actualmente como dan conocer los cursos y actividades que como emprendedores ofrecen?
- Si de último momento se pospone, como le notifican a las personas que ya pertenecen a emprendedores. ?
- ¿Actualmente como una persona se suscribe a Emprendedores Hosanna?
- ¿Cómo se clasifican los cursos que imparte esta asociación?
- ¿Qué información les gustaría que se solicitara en la aplicación móvil para la suscripción y con qué objetivo?
- ¿Las imágenes que se imprimen en los boletines son almacenada en alguna base de datos?

FORMATO DE ENCUESTA

Universidad Nacional Autónoma de Nicaragua
Unan- Managua
RURD

Encuesta a l@s miembros de la Iglesia Hosanna del departamento de Managua.

Estimados miembros: La presente encuesta debe llenarse en todas sus preguntas con veracidad en las respuestas. Todos los datos que aporten, serán de mucho valor para nuestro trabajo de investigación:” Desarrollo de Aplicación móvil en Android para....”. Cualquier aclaración de alguna pregunta, hágasela al encuestador. Gracias por su colaboración.

Edad: _____

Sexo:

Masculino____ Femenino _____

Marque con una x la opción de su preferencia.

1. ¿Posees un Smartphone?
 Si
 No

2. ¿Qué Sistema Operativo utiliza su Smartphone?
 Android
 Windows Phone
 Otro: _____

3. Utiliza usted aplicaciones móviles en su celular?
 Si
 No

4. Tiene usted conocimiento de la Asociación de Emprendedores Hosanna?
 Si
 No

5. Crees que es muy limitado las alternativas de consulta de actividades o suscripción que te brinda la Asociación de Emprendedores?
- Si
 - No
6. Como calificarías las alternativas de tener una aplicación móvil, gratuita y fácil de utilizar que le permita consultar los cursos y actividades que realiza Asociación de Emprendedores Hosanna en tu móvil?
- Buena
 - Regular
 - Mala
 - Te da igual
7. ¿Una vez desarrollada la aplicación móvil para suscripción y notificaciones de Asociación Emprendedores Hosanna, usted la utilizaría para informarse sobre cursos que ofrece y actividades que realiza dicha asociación?
- Si.
 - No.

BOLETIN SEMANAL DE COMUNIDAD HOSANNA

HOSANNA | Asociación de Emprendedores

PRESENTAN:

Ruta hacia la LIBERTAD FINANCIERA

En compañía del PASTOR EDWIN CASTRO

Viernes 29 de Agosto | **Sábado 30 de Agosto**

¡ADQUIERE TU TICKET!

MINISTERIO DE BAUTISMOS
Amuncia clases de Bautismo
 17 y 24 de Agosto
 Hora: 10:00 am
 Lugar: Salón de Conferencias

Día del Bautismo
 Sábado: 30 de Agosto
 Hora: 04:00 pm
 Lugar: Salón de Conferencias

ESCUELA DE CREENTES
SÁBADO
 30 de Agosto
 Hora: 09:00 am
 Lugar: Salón de Conferencias

DOMINGO
 31 de Agosto
 Hora: 09:00 am
 Lugar: Salón de Conferencias

RETIRO DE CANTOS
30 y 31 DE AGOSTO
 Hora: 08:00 am
 Lugar: Salón de Conferencias

CCLH

VIERNES 22 DE AGOSTO

¡ADQUIERE TU TICKET!