



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

**DESARROLLO DE UNA APLICACIÓN PARA DISPOSITIVOS MÓVILES CON
SISTEMA OPERATIVO ANDROID APLICANDO EXTREME PROGRAMMING**

Trabajo de Titulación presentado en conformidad a los requisitos
establecidos para optar por el título de:
Ingeniero en Sistemas de Computación e Informática

Profesor Guía:
Ing. Marco Galarza C.

Autor:
Erik David Mafla Narváez

Año
2012

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el/la estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Marco Galarza C.
Ingeniero en Sistemas
C.I.: 70277325-0

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

Erik David Mafla Narváez

C.I.: 171755183-0

AGRADECIMIENTOS

Culminar esta etapa de mi vida ha sido una de mis grandes metas, la cual he podido compartir en el transcurso del tiempo con mis amigos y familia

Gracias a Dios que ha sido mi guía y su palabra me ha permitido sobrepasar todo y seguir adelante

Agradezco a mi padre porque gracias a su formación soy un hombre dedicado, apasionado, disciplinado y luchador, sé que estaría orgulloso de mí. Gracias a mi madre por su fortaleza y su motivación para seguir adelante, sin ella mi formación académica no hubiese sido posible, mi madre es un ejemplo a seguir.

Son tantas las personas que han aportado para que esto sea posible, mis mejores amigos, la familia y la universidad, que me formó con los mejores profesores y sus conocimientos me han permitido ser excelente un profesional.

DEDICATORIA

A Dios por ser mi fortaleza y guía.

A mi padre, que a pesar de su ausencia puedo sentir su apoyo y presencia.

A mi madre por su esfuerzo y ayuda incondicional, pero sobre todo por su ejemplo de lucha, superación y fe.

RESUMEN

El presente trabajo de tesis se realizó con el propósito de solucionar las necesidades de la empresa Renase Cia. Ltda., creando una aplicación móvil para sistemas operativos Android y utilizando la metodología Extream Programing.

Para el desarrollo de este proyecto se realizó un análisis de la metodología Extream Programing, el sistema operativo Android y de las herramientas de desarrollo Java y MySql.

Antes del desarrollo del sistema y aplicando la metodología mencionada, se realizó el análisis y diseño de la aplicación, en donde se identificó el problema, se detalló los requerimientos de la empresa, se analizó la plataforma, se diseño la base de datos, se creo los escenarios y tarjetas CRC y finalmente se definió la metáfora que se uso para el desarrollo del sistema.

El sistema realizado despliega la lista de productos de la empresa y sus características como precio, stock, descripción, presentación e imágenes y permite realizar una búsqueda del producto a través de su nombre. Además, la aplicación registra al cliente en el sistema para posteriormente obtener la afiliación y que éste pueda realizar los pedidos; también permite la actualización de su información y el despliegue de lo pedidos realizados. El cliente puede seleccionar los productos deseados y agregarlos a un carrito de compras, el mismo que al finalizar se guardará en una base de datos MySql instalada en el servidor principal de la empresa, para que posteriormente el sistema de la empresa lo pueda consultar y pueda ser despachado.

Con esta aplicación la empresa Renase tendrá la posibilidad de llevar un sistema de control ordenado de sus ventas, productos y clientes; logrando satisfacer de mejor manera a sus clientes actuales y futuros, entregando pedidos a tiempo y con un mejor control internamente.

SUMMARY

This thesis work was done in order to meet the needs of the company Renase Cia. Ltda., creating a mobile app for Android operating systems using the methodology Extream Programing.

For the development of this project, there is an analysis of the methodology Extream Programin, the Android operating system, Java and MySql development tools.

Before the development of the system and applying the methodology mentioned above was carried out the analysis and design of the application, where the problem was identified, detailed business requirements, it analyzed the platform, database design, it created the CRC cards and scenarios and finally defined the metaphor that was used for system development.

The system displays the list made the company's products and their characteristics such as price, stock, description, presentation and images and lets you search by product name. In addition, the application logs the customer into the system for membership and then get him to carry out orders, but also allows for updating their information and the deployment of the orders. The customer can select the desired products and add them to a shopping cart. The same as the end will be saved in a MySql database installed on the server of the company and then are obtained by the enterprise system and can be dispatched.

With this application the company Renase will be able to carry a set of ordered control of sales, products and customers, achieving a better way to satisfy their current and future customers, delivering orders on time and with better control internally.

ÍNDICE

INTRODUCCIÓN	1
1 CAPÍTULO I. METODOLOGÍA EXTREAM PROGRAMING (XP)	3
1.1 INTRODUCCIÓN.....	3
1.2 EXTREAM PROGRAMING (XP)	4
1.3 CARACTERÍSTICAS DE XP	5
1.3.1 Pruebas Unitarias	5
1.3.2 Refabricación	6
1.3.3 Programación en Pares	6
1.4 VALORES DE XP	6
1.4.1 Comunicación	6
1.4.2 Simplicidad	7
1.4.3 Retroalimentación	7
1.5 PROPUESTA DE XP	8
1.5.1 Incrementos	8
1.5.2 Bienvenidos los Cambios.....	9
1.5.3 Costeo Estándar de Cambios	9
1.5.4 Avanzar por lo Requerido	9
1.5.5 El Cliente como Miembro del Equipo	10
1.6 ROLES EN XP.....	10
1.7 REGLAS Y PRACTICAS DE XP	11
1.7.1 Planificación.....	11
1.7.2 Diseño.....	12
1.7.3 Codificación	13
1.7.4 Pruebas (Test)	14
1.8 CICLO DE VIDA Y FASES DE XP	15
1.8.1 Exploración	15
1.8.2 Planificación de Entregas	16
1.8.3 Iteraciones	16
1.8.4 Producción.....	17
1.8.5 Mantenimiento	17
1.8.6 Muerte del Proyecto.....	18
2 CAPÍTULO II. SISTEMA OPERATIVO ANDROID	19
2.1 INTRODUCCIÓN.....	19
2.2 HISTORIA Y POSICIONAMIENTO	19
2.3 CARACTERÍSTICAS	21
2.4 ARQUITECTURA	23
2.5 VERSIONES	26
2.6 DESARROLLO ANDROID	37
2.6.1 SDK de Android	37

2.6.2	Arquitectura de las Aplicaciones	39
2.6.3	Anatomía de una Aplicación	40
2.6.4	Interfaz de Usuario	43
3	CAPÍTULO III. HERRAMIENTAS DE DESARROLLO	45
3.1	MOTODEV STUDIO FOR ANDROID	45
3.2	ADT PLUGIN DE ANDROID	48
4	CAPÍTULO IV. ANÁLISIS Y DISEÑO DE LA APLICACIÓN	49
4.1	ANTECEDENTES	49
4.2	IDENTIFICACIÓN DEL PROBLEMA	49
4.3	HISTORIAS DE USUARIO	51
4.4	PLAN DE ENTREGAS	57
4.5	ITERACIONES	61
4.6	CREACIÓN DE ESCENARIOS Y DE TARJETAS CRC	63
4.7	DEFINICIÓN DE LA METÁFORA DEL SISTEMA	68
4.7.1	Definición de la Metáfora en los Objetos	69
4.7.2	Definición de la Metáfora en los Componentes	69
4.7.3	Definición de la Metáfora en las Variables y Métodos	70
4.8	DISEÑO DE LA APLICACIÓN	70
4.8.1	Arquitectura de la Aplicación	70
4.8.2	Diseño de la Base de Datos	72
5	CAPÍTULO V. IMPLEMENTACIÓN Y PRUEBAS	74
5.1	IMPLEMENTACIÓN	74
5.2	PRUEBAS DE UNIDAD Y ACEPTACIÓN	75
5.3	DESARROLLO DE ITERACIONES	77
5.4	DOCUMENTACIÓN DE LA APLICACIÓN	79
5.4.1	Plataforma	79
6	CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES	81
6.1	CONCLUSIONES	81
6.2	RECOMENDACIONES	83
	Referencias	85
	Glosario	87
	Anexos	88

INTRODUCCIÓN

El presente proyecto de tesis se refiere al desarrollo de un sistema para dispositivos móviles Android usando la metodología de desarrollo Extream Programing, el cual será usado por la empresa Renase, dedicada a la elaboración de medicina natural, para llevar un mejor control de sus ventas y de sus clientes.

En la actualidad el avance de la ciencia y la tecnología han impulsado al ser humano a explorar nuevos servicios, los cuales ayudan en el ahorro de tiempo y dinero. Por esta razón es de interés el estudio de nuevas tecnologías que aportan con soluciones rápidas y eficaces, y que permiten al ser humano satisfacer sus necesidades cada vez más exigentes. Entre estas nuevas tecnologías se encuentra el rápido desarrollo y avance de los dispositivos móviles, que brindan comodidad, rapidez, interfaces más amigables que las PC's, posibilidad de conexión a redes inalámbricas de datos, posicionamiento global (GPS), comercio electrónico, entre otras soluciones a medida.

El sistema operativo Android es uno de los más vendidos y usado en dispositivos móviles como celulares, tabletas y computadores, por su adaptabilidad a estas terminales, miles de aplicaciones y su desarrollo y crecimiento constante.

Actualmente la empresa Renase no lleva un control sistematizado de sus ventas, clientes y productos, por lo cual, se ha pedido realizar un aplicativo móvil sobre el sistema operativo Android, el cual permita mostrar la información actualizada de todos sus productos, registrar sus actuales y futuros clientes y realizar pedidos. El aplicativo guardará toda esta información en una base de datos que posteriormente será utilizada por su sistema interno.

Este proyecto abarca los siguientes temas: Metodología Extream Programming, Sistema Operativo Android y sus Herramientas para el desarrollo, Análisis, Diseño, Implementación y Pruebas de la Aplicación.

Finalmente se indicaran las conclusiones y recomendaciones obtenidas luego de la culminación del proyecto.

1 CAPÍTULO I. METODOLOGÍA EXTREAM PROGRAMING (XP)

1.1 INTRODUCCIÓN

En la actualidad el desarrollo de software ha requerido del uso de diferentes metodologías de desarrollo que ayuden en el proceso del desarrollo. Es por esta razón que han nacido diferentes metodologías las cuales inciden en distintas dimensiones de este proceso. Entre estas se encuentran las metodologías ágiles, que dan mayor valor al individuo, a la relación con el cliente y al desarrollo incremental con iteraciones cortas. Las metodologías ágiles se basan en las metodologías tradicionales y toman en cuenta lo más importante de cada una de estas, dejando a un lado el manejo excesivo de documentación y burocracia.

José H. Canós, Patricio Letelier y Ma Carmen Penadés (2004) en su libro “Metodologías Ágiles en el Desarrollo de Software” detallan en tabla 1 las principales diferencias entre las metodologías ágiles y tradicionales.

Tabla 1 Comparación entre metodologías ágiles y tradicionales

DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES	
Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<20 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tomado de: José H. Canós, Patricio Letelier y Ma Carmen Penadés (2004)

Las metodologías tradicionales han sido la pauta para la creación de las metodologías ágiles, que nacieron con el propósito de mejorar y hacer más eficiente el desarrollo de proyectos pequeños y medianos, permitiendo el ahorro de tiempo, recursos humanos, materiales y mayor flexibilidad en la aceptación de cambios e innovaciones.

1.2 EXTREAM PROGRAMING (XP)

En este proyecto se utilizará la metodología de desarrollo ágil Extream Programing (Programación Extrema) o XP como se la llamara de aquí en adelante.

XP fue creada por Kent Beck en 1996 y se basa en la simplicidad, la comunicación, la retroalimentación continua y la reutilización del código desarrollado. También es una metodología de desarrollo bastante flexible, ya que es efectiva cuando los requisitos que establece el cliente para el desarrollo de software están en constante cambio. Esta metodología propone entregar el software que los clientes necesitan en el momento en que lo requieran, respondiendo a los requisitos cambiantes de los clientes, aún en fases tardías de ciclo de vida del desarrollo. (José Carlos Cortizo Pérez, Diego Expósito Gil y Miguel Ruiz Leyva, 2011)

1.3 CARACTERÍSTICAS DE XP

Las características más importantes de XP son; las pruebas unitarias, refabricación y la programación en pares, las cuales hacen que XP sea un método de desarrollo ágil.

1.3.1 Pruebas Unitarias

Se basa en realizar un completo conjunto de pruebas a los principales procesos o módulos del sistema incluso antes de terminarlos, de tal manera que se previene problemas o errores que puedan llegar a ocurrir. También estas pruebas ayudan a evitar errores en puntos del proyecto en que sería muy costoso llegar a corregirlas; por ejemplo, al finalizar el ciclo de vida del proyecto.

Las pruebas continuas sobre el código ayudan a que éste sea más fiable y se tenga estabilidad en el trabajo de programación realizado. Por ejemplo, al tener un modulo del sistema con código y funcionamiento probado se tendrá la seguridad de reutilizar este código en otros módulos e inclusive en otros sistemas.

1.3.2 Refabricación

La refabricación es el proceso de rediseño del código después de haberlo escrito y se basa en el mejoramiento continuo del código, eliminando la duplicación y aumentando la cohesión del código, al tiempo que reduce el acoplamiento (el aumento de la cohesión y el bajo tiempo de acoplamiento se ha reconocido como las características de un buen diseño de código). Esto resulta en un código simple con un buen diseño y de fácil lectura y comprensión. (Jeffries, 2001)

Una de las ventajas de la refabricación es la eliminación del código duplicado que es una clara señal de un mal diseño. Por ejemplo, el código duplicado ensucia el programa haciéndolo mas largo, aumentando la posibilidad de obtener errores y haciéndolo incomprensible.

1.3.3 Programación en Pares

La programación en pares se basa en que dos desarrolladores miembros del proyecto participan en una misma estación de trabajo, y que cada miembro realiza la acción que el otro no está haciendo en ese momento. Por ejemplo, mientras un desarrollador esta escribiendo el código, el otro lo revisa con la intención de detectar errores en el proceso, ahorrar tiempo y esfuerzo, lo que se pierde cuando un solo desarrollador cumple con esta tarea.

1.4 VALORES DE XP

Los principales valores en los que XP se fundamenta son la comunicación, simplicidad y la retroalimentación.

1.4.1 Comunicación

Se refiere a la constante comunicación del equipo de trabajo incluyendo al cliente, logrando así eliminar una gran cantidad de pasos intermedios utilizados

en las metodologías clásicas, como por ejemplo, la documentación que tomaba mucho tiempo y esfuerzo.

En XP es necesario integrar al usuario final como parte del equipo, ya que así se obtiene una retroalimentación real del proyecto que permiten a los desarrolladores identificar más fácilmente los requerimientos, y al cliente sentir el avance del proyecto.

1.4.2 Simplicidad

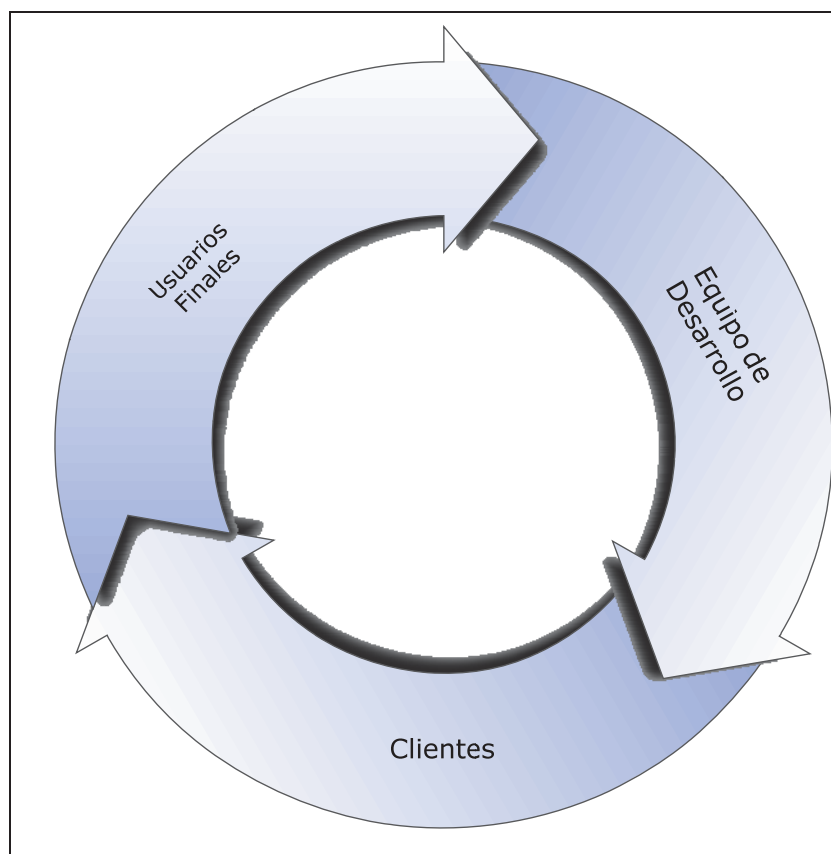
La simplicidad es una de las bases en XP y que siempre debe estar presente en todas las fases del proyecto. Se caracteriza por:

- Comenzar por los requerimientos más sencillos.
- Preferir siempre el desarrollo sencillo a una interface complicada.
- Si algo no se requiere al momento se descarta, asumiendo que será requerido después.
- No se incluye cosas adicionales que no se han requerido.

1.4.3 Retroalimentación

Se basa en el continuo contacto con el equipo de desarrollo, el cliente y el usuario final. Así se puede tener una valoración constante del avance del proyecto y que el cliente y los usuarios finales informen constantemente al equipo lo que realmente desean.

Gráfico 1 Retroalimentación en XP



Elaborado por: El autor

La aplicación de estos tres valores ayuda al desarrollo del proyecto permitiendo un mejor trabajo en equipo, y proyectando la eficiencia, calidad y sencillez en la elaboración del mismo.

1.5 PROPUESTA DE XP

La propuesta de XP se deriva en cinco ámbitos: incrementos, bienvenidos los cambios, costeo estándar de cambios, avanzar por lo requerido y el cliente como miembro del equipo.

1.5.1 Incrementos

En XP se realizan incrementos de software que son los entregables de cada iteración (ciclos de desarrollo cortos), los cuales se realizan a lo largo del

proyecto. Las iteraciones comienzan en pequeño y añaden funcionalidades con retroalimentación continua, lo que significa que el cliente y el usuario final están siempre al tanto de cada iteración, y con la retroalimentación se genera cambios que son los incrementos de software.

1.5.2 Bienvenidos los Cambios

En XP los cambios son bienvenidos en cualquier etapa del proyecto, lo que implica que el equipo de desarrollo debe ser lo suficientemente flexible y dispuesto al cambio. La refabricación es una de las características de XP que ayuda a aplicar estos cambios y a no rechazarlos como en otras metodologías, donde el cambio en una fase avanzada del proyecto podría llegar a ser muy costoso.

1.5.3 Costeo Estándar de Cambios

El objetivo de XP es que los costos de los cambios se reduzcan usando sus características, valores y buenas prácticas de desarrollo, las cuales harán que el proyecto sea más flexible a los cambios. Las pruebas en cada versión liberada es una de las características de XP que hacen que el costo de los cambios sea mucho más barato que en otras metodologías de desarrollo, que primero realizan el análisis, después el diseño, implementación, pruebas y producción, mientras que en XP siempre se esta implementando, probando y produciendo.

1.5.4 Avanzar por lo Requerido

XP propone no incluir funcionalidades a menos que sea necesario. Las funcionalidades propuestas deberán ser analizadas por el equipo a fin de seleccionar únicamente las que cumplen con las necesidades del cliente, las que no fueron seleccionadas serán guardadas para en un futuro si se requiere implementarlas.

1.5.5 El Cliente como Miembro del Equipo

El cliente es uno de los miembros más importantes del equipo en XP, él cual debe estar informado del proyecto e implicado en el proceso de desarrollo. También es quien aclarará las dudas del equipo en el proceso.

1.6 ROLES EN XP

En la tabla 2 se encuentra la descripción de los roles que existen en XP.

Tabla 2 Roles de XP (José H. Canós, Patricio Letelier y Ma Carmen Penadés, 2004)

Roles en la XP de acuerdo a lo propuesto por Beck	
Roles	Funciones
Programador	Escribe la pruebas y genera el código.
Cliente	Escribe las Historias de Usuario, elabora pruebas funcionales y esta en constante comunicación con el equipo.
Encargado de las Pruebas (Tester)	Escribe las pruebas junto con el cliente y el programador, las ejecuta y difunde los resultados.
Encargado de seguimiento (Tracker)	Encargado de la evolución del proyecto y corregir errores.
Entrenador (Coach)	Dirige el proyecto y verifica procesos.
Consultor	Miembro externo especialista en un determinado tema necesario para el proyecto.
Gestor (Big boss)	Coordina y supervisa el proyecto.

Elaborado por: El autor

Es importante definir los roles de cada integrante antes de comenzar el proyecto; de ésta manera el equipo se centrará en su función, siendo el trabajo eficiente y evitando sobrecarga de los recursos.

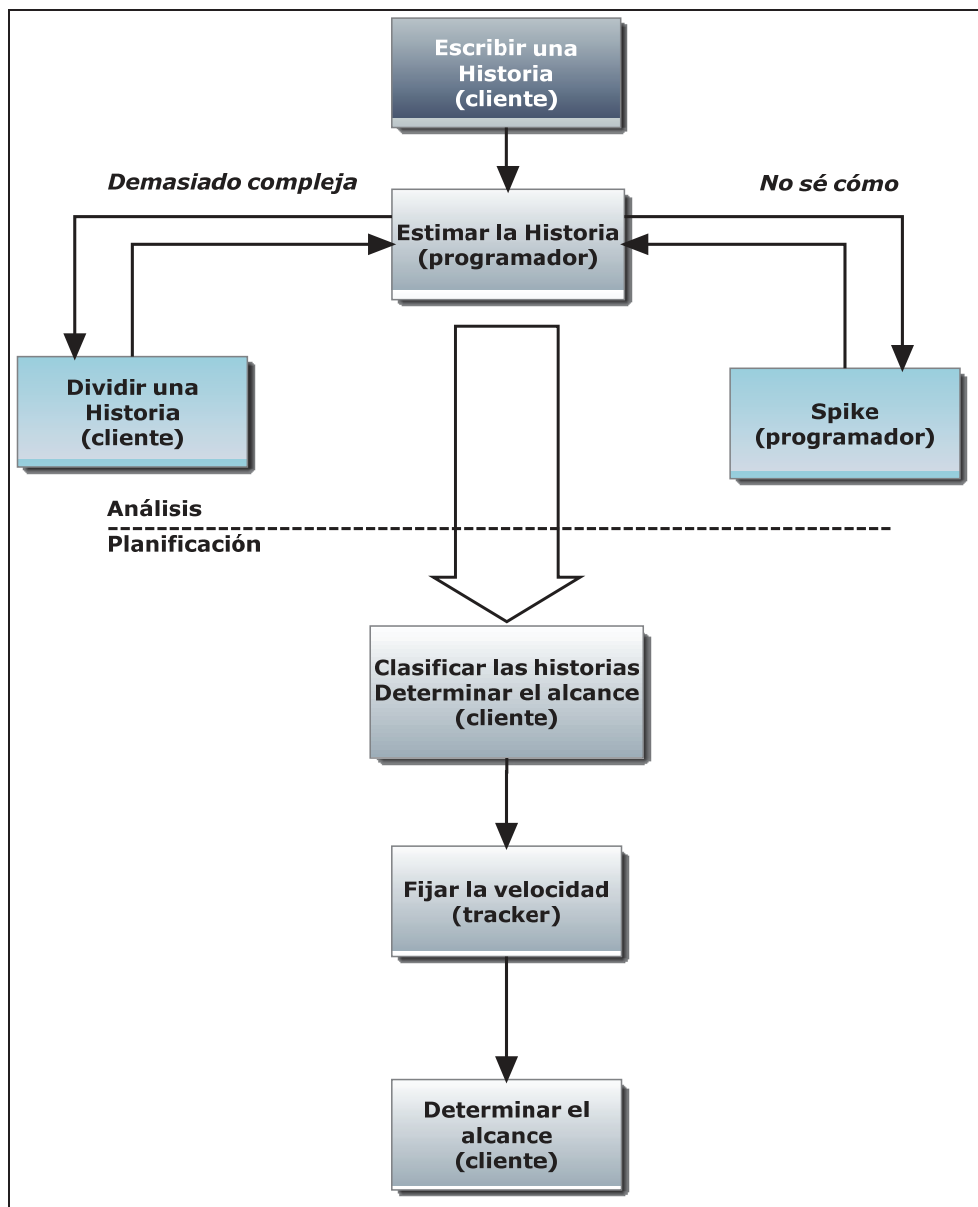
1.7 REGLAS Y PRACTICAS DE XP

XP define cuatro reglas y practicas importantes que guían los diferentes aspectos del desarrollo. Estas reglas y prácticas se dividen en las siguientes áreas del desarrollo: Planificación, Diseño, Codificación y Pruebas.

1.7.1 Planificación

La planificación comienza recopilando las **historias de usuario (user stories)** que son documentos generados por el cliente, los cuales contienen descripciones cortas de lo que el sistema debe realizar. Las historias de usuario sustituyen a los casos de uso y son analizadas por los desarrolladores quienes evaluarán el tiempo que tomará desarrollar cada una de estas. Si alguna historia de usuario presenta dificultades para determinar su complejidad, se desarrollan pequeños programas de pruebas llamados **spikes** para así poder determinar el tiempo de desarrollo o en casos extremos se pedirá al cliente dividir la historia de usuario para que esta sea más sencilla. Una vez obtenida las estimaciones de tiempo se organiza una reunión de planificación con el cliente, los desarrolladores y el usuario final en donde se desarrolla un plan o cronograma llamado **plan de entregas (“Release Plan”)**. Una vez obtenido el plan de entregas, se comienza una fase de iteraciones en donde se desarrolla, prueba e instala unas pocas Historias de Usuario definidas por los desarrolladores y el cliente para determinar la velocidad de entrega. Posteriormente se organiza una nueva reunión para evaluar el plan de entrega y si es necesario modificarlo.

Gráfico 2 Planificación en XP



Elaborado por: El autor

1.7.2 Diseño

En XP el diseño debe cumplir con las siguientes características:

- Contar con una arquitectura base sobre la cual se desarrollara el sistema.
- Los diseños deben ser simples y claros, los cuales son más rápidos de implementar.

- Elegir una metáfora de sistemas que ayude a manejar un mismo idioma al momento de desarrollar, y de esta manera sea sencillo explicar el propósito del proyecto y guiar la estructura y arquitectura del mismo.
- Las tarjetas CRC (Clase, Responsabilidad y Colaboración) se utilizan individualmente para representar objetos. La clase del objeto pueden ser escritas en la parte superior de la tarjeta, las responsabilidades enumeradas en la parte izquierda y las clases que colaboran son listadas a la derecha de cada responsabilidad.
- Crear soluciones óptimas para que sea más sencillo averiguar las dificultades técnicas o problemas en el diseño.
- Concentrarse en añadir las funcionalidades requeridas dejando las necesidades futuras pendientes.
- Reutilizar el código cada vez que sea necesario y siempre y cuando sea posible.

1.7.3 Codificación

En la codificación o desarrollo deben estar presentes los siguientes aspectos:

- “El cliente siempre tiene que estar disponible” es uno de los requerimientos de XP, no solo como apoyo sino como parte del equipo. Es necesaria una comunicación directa con el cliente en toda la fase de desarrollo, de preferencia cara a cara y en el mismo sitio.
- El código debe ser escrito de manera que cumpla con los estándares definidos por el equipo al comienzo del proyecto, y de esta manera sea fácil de entenderlo y de recodificarlo.

- La Programación dirigida por las pruebas (Test-driven programming), es decir, lo primero que se escribe son las pruebas unitarias que cada modulo debe cumplir al finalizar su desarrollo.
- La programación en pares, la cual aumenta la calidad del software sin impactar en el tiempo de entrega.
- Las integraciones permanentes ayudan a que siempre el equipo trabaje sobre la última versión, lo cual evita que se hagan cambios o mejoras sobre versiones antiguas que causarían graves problemas de integración.
- La propiedad colectiva del código significa que el equipo de desarrollo puede contribuir con ideas que ayuden a mejorar cualquier parte del código; así también, cualquier pareja puede cambiar el código que sea necesario para solucionar problemas, agregar funcionalidades o recodificar.
- La metodología XP indica que se debe llevar un ritmo sostenido de trabajo, sin importar las horas que se trabajen por semana. El concepto es establecer una práctica de trabajo planificada que sea de manera constante y razonable, sin sobrecargar al equipo o retrasar el proyecto.

1.7.4 Pruebas (Test)

En esta etapa se debe tomar en cuenta los siguientes detalles:

- Todo el código debe tener un plan de pruebas unitarias incluso antes de comenzar la programación. Los módulos desarrollados deben cumplir con estas pruebas antes de ser liberados o publicados.
- Se debe corregir inmediatamente el código cuando se encuentre un bug (error), volviendo a generar pruebas que certifiquen que el error haya sido resuelto.

- Se prueba la tecnología y se exploran las soluciones, construyendo un prototipo del sistema.

Esta fase puede tardar varias semanas dependiendo de tamaño del proyecto y la familiaridad que tengan los programadores con la tecnología. (Patricio Letelier y M^a Carmen Penadés, 2010)

1.8.2 Planificación de Entregas

Esta es la fase más importante ya que aquí se marca el inicio del proyecto y el proceso que se llevará a cabo hasta su culminación. A continuación se describe los pasos a seguir en esta fase:

- Se reúne el cliente con el equipo y se establece la prioridad de cada historia de usuario.
- EL equipo de desarrollo dará una estimación de tiempo y de esfuerzo que tome cada historia de usuario.
- Conjuntamente con el cliente se toman acuerdos sobre el contenido de la primera entrega y se realiza un cronograma o plan de entregas. El tiempo que debe tomar cada entregable no debe exceder de tres meses.

El tiempo que toma esta fase es de pocos días. (Patricio Letelier y M^a Carmen Penadés, 2010)

1.8.3 Iteraciones

Las iteraciones son ciclos de desarrollos cortos, con entregables funcionales al finalizar cada ciclo. Cada iteración realiza un proceso completo de análisis, diseño, desarrollo y pruebas utilizando las reglas y practicas de la metodología XP.

En esta fase se desarrollan las funcionalidades del sistema y se establecen varias iteraciones antes de que el sistema sea entregado. También se constituye un plan de iteraciones que debe tomar en cuenta los siguientes elementos:

- Historias de usuario no abordadas.
- Velocidad del proyecto.
- Pruebas de aceptación no superadas en iteraciones anteriores.
- Tareas no terminadas en iteraciones anteriores.

Cada iteración no debe llevar más de tres semanas y debe ser asignada a un programador responsable, pero como indica XP, será llevada a cabo por parejas de programadores. (Patricio Letelier y M^a Carmen Penadés, 2010)

1.8.4 Producción

En esta fase se debe realizar pruebas funcionales y de rendimiento antes de poner el sistema en ambiente de producción e incluso el cliente puede solicitar algún cambio o afinación en el sistema.

Una vez realizadas las pruebas adicionales y ajustes en el sistema la primera versión del sistema entra en producción.

En esta fase también se puede analizar el aumento de nuevas funcionalidades o características las cuales serán documentarlas para posteriormente agregarlas en el sistema. (Joskowicz, 2008)

1.8.5 Mantenimiento

En esta fase la primera versión del proyecto ya se encuentra en producción y es necesario un equipo de soporte que trabaje conjuntamente con el cliente; por otro lado, se trabaja con el equipo de desarrollo elaborando nuevas

iteraciones para continuar con el desarrollo. (Patricio Letelier y M^a Carmen Penadés, 2010)

1.8.6 Muerte del Proyecto

Esta fase del proyecto todas las historias de usuario ya están desarrolladas e incluidas en el sistema. El cliente no requiere que se incluyan más funcionalidades y el sistema desarrollado cumple con las necesidades del cliente. Se genera la documentación final y ya no se realizan cambios en el sistema.

También se puede dar por terminado el proyecto cuando el sistema no cumple con lo requerido y esperado por el cliente o cuando ya no existe presupuesto para continuar con el mismo. (Patricio Letelier y M^a Carmen Penadés, 2010)

2 CAPÍTULO II. SISTEMA OPERATIVO ANDROID

2.1 INTRODUCCIÓN

Android es un sistema operativo de código abierto basado en el núcleo Linux, diseñado y desarrollado para dispositivos móviles por la empresa Android Inc., una firma comprada por Google. Posteriormente fue desarrollado para soportar otros dispositivos como tablets, netbooks, reproductores mp3, PC's, televisores, entre otros que con el tiempo han ido adoptando este sistema operativo.

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones, las cuales se pueden descargar desde su tienda oficial Google Play anteriormente conocida como Google Market y que en la actualidad supera las 400.000 aplicaciones sin contar miles aplicaciones que se pueden descargar desde sitios no oficiales. (Wikipedia, 2011)

Gráfico 4 Logo oficial del sistema operativo Android



Tomado de: http://en.wikipedia.org/wiki/Android_version_history

2.2 HISTORIA Y POSICIONAMIENTO

Android fue desarrollado por Android Inc. una pequeña compañía comprada en julio del 2005 por la empresa Google. Actualmente Android es desarrollado por la Open Handset Alliance, un consorcio de empresas fabricantes,

desarrolladoras de hardware, software y operadores de servicio lideradas por Google.

Los inicios de Android fueron un poco lentos debido a que primero se lanzó el sistema operativo y su SDK (software development kit) antes de su primer móvil el HTC DREAM, conocido más como T-Mobile G1 y lanzado el 22 de octubre del 2008. Android rápidamente se posicionó en los primeros lugares de los sistemas operativos móviles más vendidos en el mundo y actualmente posee el 38,5% de cuota en la escala mundial de teléfonos móviles. (Wikipedia, 2011)

En la tabla 3 se indica la cuota de los sistemas operativos móviles más vendidos por año.

Tabla 3 Cuota de los sistema operativos móviles mas vendidos. (Wikipedia, 2011)

Sistema operativo	Años		
	2011	2010	2009
Android	38,50%	25,50%	3,50%
iOS	19,40%	16,70%	17,10%
Symbian OS	19,20%	36,60%	44,60%
BlackBerry OS	13,40%	14,80%	20,70%
Windows Phone	5,50%		
Linux embebido		2,10%	4,70%
MeeGo			
Bada			
Maemo			
Java ME			
Windows CE			
Otros	3,90%	1,50%	1,50%

Elaborado por: El autor

2.3 CARACTERÍSTICAS

A continuación se describen las principales características del sistema operativo Android:

- **Es abierto.**- lo cual permite acceder a todas las APIs del bajo nivel del equipo.
- **Eliminación de fronteras.**- permite combinar la información del dispositivo con la información de la web para generar nuevas experiencias.
- **Igualdad.**- no existen diferencias entre las aplicaciones de terceros y las propias del sistema.
- **Flexibilidad.**- se puede utilizar y remplazar cualquier componente fácilmente.
- **Seguridad.**- cada aplicación elige que permisos de acceso requiere.
- **Layout.**- es adaptable a cualquier tipo y resolución de pantalla y posee bibliotecas de gráficos 2D y 3D basada en las especificaciones de OpenGL ES 2.0.
- **Almacenamiento.**- utiliza SQLite, una base de datos liviana para almacenamiento de datos estructurado, con soporte de SQL e integración directa con las aplicaciones.
- **Conectividad.**- el sistema operativo soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE y WiMAX (El uso de estas depende de la terminal).

- **Mensajería.-** soporta los servicios SMS, MMS y mensajes de hilo.
- **Navegador Web.-** Android contiene un navegador web basado en el motor WebKit (igual que safari) y soporta Ajax.
- **Soporte de Java.-** en Android no existe una máquina virtual Java, pese a que sus aplicaciones están escritas en ese lenguaje; en lugar de esta, corre su propia máquina Virtual Dalvik.
- **Multimedia.-** el sistema soporta varios formatos multimedia de audio, video e imagen, convirtiéndolo en un perfecto y muy completo reproductor multimedia. La reproducción de los formatos depende del dispositivo y la aplicación.
- **Soporte para Streaming.-** soporta Streaming como RTP/RTSP y descarga progresiva de HTML (HTML5 <video> tag). También dependiendo del dispositivo soporta Adobe Flash Streaming (RTMP) utilizando una aplicación en el sistema de Adobe Flash Player.
- **Soporte para Hardware adicional.-** soporta una diversidad de dispositivos. Entre los mas usados están las cámaras de fotos y vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, sensores de proximidad y de presión, y aceleración 2D y 3D. Existe en el mercado una diversidad de dispositivos que incluyen mejoras y nueva tecnología en hardware a las cuales Android se adapta perfectamente.
- **Entorno de desarrollo.-** Android ofrece un entorno de desarrollo multiplataforma gracias a su SDK que corre en diferentes IDE's de desarrollo Java populares como Eclipse o Netbeans, por medio de un plugin para el desarrollo en Android. Ofrece también por medio de su SDK la posibilidad de descargar todas las versiones de Android y emularlas en una pantalla útil para realizar pruebas de desarrollo.

- **Multi-táctil.**- incluye un soporte nativo para pantallas multi-táctiles.
- **Bluetooth.**- soporta A2DF y AVRCP, envío de archivos y exploración del directorio telefónico.
- **Video llamada.**- Android soporta video llamada a través de diversas aplicaciones como Skype, Fring o Google Talk.
- **Multitarea.**- permite la ejecución de tareas en segundo plano.
- **Características basadas en voz.**- permite búsqueda en Google por medio de voz.
- **Tethering.**- permite que el teléfono sea usado como punto de acceso alámbrico o inalámbrico a internet desde la PC usando conexión 3G. (Wikipedia, 2011; Mundo Manuales, 2011)

Todas estas características han hecho que Android sea en este tiempo uno de los sistemas operativos más usados en dispositivos móviles y tablets.

Varias compañías de tecnología se han centrado en desarrollar nuevos dispositivos con mejores características, sin la necesidad de preocuparse por el sistema operativo que usaran ya que Android y sus características cumplen con los requisitos necesarios para su correcto funcionamiento.

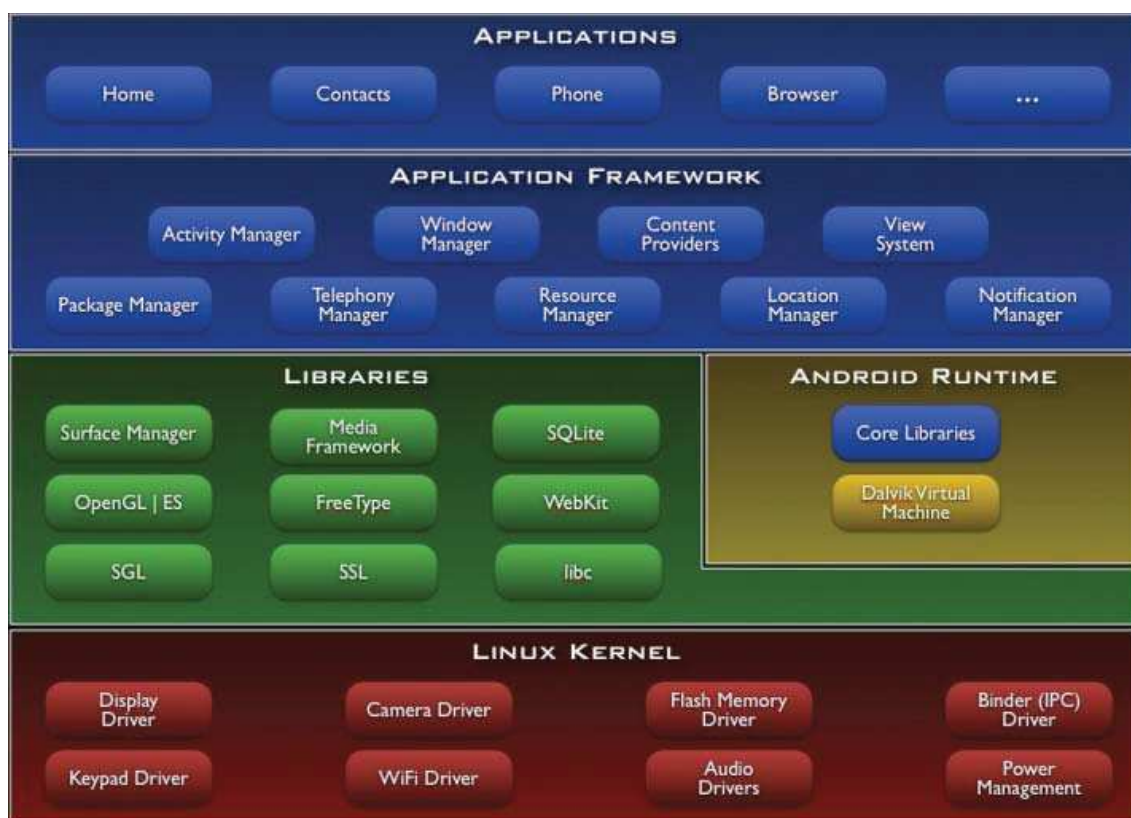
2.4 ARQUITECTURA

La arquitectura del sistema operativo Android se compone por varias capas que facilitan el desarrollo y la creación de aplicaciones. La distribución de las capas se conoce como pila, es decir, cada capa utiliza los elementos de su capa inferior para realizar sus funciones. Esto ayuda a los desarrolladores para que

no tengan que programar a bajo nivel y obtenga acceso a los diferentes componentes de hardware del dispositivo o terminal. (Androideity, 2011)

El orden de estas capas se muestra en la Gráfico 5:

Gráfico 5 Arquitectura del sistema operativo Android



Elaborado por: El autor

A continuación se detalla cada una de las capas desde la capa inferior hasta la superior:

- **Kernel Linux.-** Android está basado en Linux, específicamente en el kernel 2.6, que provee toda la gestión de drivers, memoria, seguridad y el framework de comunicaciones. (Meier, 2009)
- **Librerías o Bibliotecas.-** esta capa se compone de librerías nativas de Android, también llamadas bibliotecas, escritas en C/C++, compiladas para la arquitectura de hardware específica de teléfono y usadas por

varios componentes del sistema. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma más eficiente. (Meier, 2009)

Algunas de las librerías que encuentran habitualmente son: OpenGL (motor gráfico), Bibliotecas multimedia (formatos de audio, imagen y video), Webkit (navegador), SSL (Server Socket Logger usado para las comunicaciones), FreeType (fuentes de texto), SQLite (base de datos), entre otras.

- **Runtime de Android o entorno de ejecución.-** el entorno de ejecución de Android no se considera una capa en sí, sino que se considera parte de las librerías porque está formado por estas. Este entorno incluye un conjunto de librerías de C/C++ que son utilizadas por diversos componentes del sistema y proporciona la mayor parte de las funciones disponibles. Las aplicaciones Android a pesar de ser escritas en Java no corren sobre la máquina virtual Java de Java ME, sino que corren bajo su propia máquina virtual Dalvik, creada para optimizar el uso de memoria y permitir que un dispositivo puede correr múltiples aplicaciones, es decir, ejecuta varias instancias de la máquina virtual simultáneamente aislando los procesos. Dalvik corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx". Cabe indicar que Dalvik es una variación de la máquina virtual de Java, por lo que no es compatible con el bytecode Java. Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión.dex que es específico para Dalvik, y por ello no se puede correr aplicaciones Java en Android ni viceversa. (Meier, 2009)
- **Framework de Aplicaciones o Marco de trabajo de aplicaciones.-** esta capa está formada por todas las clases y servicios que utilizan

directamente las aplicaciones y a las cuales los desarrolladores tienen acceso completo permitiendo la creación de aplicaciones innovadoras. La arquitectura permite la reutilización de componentes, es decir, que una aplicación puede hacer uso de otra publicada respetando las restricciones de seguridad, como el acceso a GPS, a la barra de notificación, etc. Estos componentes, son librerías JAVA, que acceden a los recursos a través de la máquina virtual Dalvik. (Meier, 2009)

- **Aplicaciones:** en esta última capa se incluyen todas las aplicaciones del dispositivo, las nativas (programadas en C o C++), las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado. Por ejemplo, algunas de las aplicaciones que vienen preinstaladas en esta capa son inicio (Home) y lanzador (Launcher), las cuales permiten ejecutar otras aplicaciones mediante una lista o mostrando diferentes escritorios donde se puede ubicar accesos directos e incluso widgets (pequeños programas que ocupan poco espacio y permiten el acceso a la información directamente sin tener que ingresar a ninguna aplicación). (Meier, 2009)

2.5 VERSIONES

Desde la salida de Android, el sistema ha sufrido constantes actualizaciones en el sistema operativo base, las cuales han ido agregando características y mejoras en el sistema.

A continuación, en orden desde su lanzamiento, se describen las versiones de Android con su respectivo código en nombre y características.

- **Android 1.0**

Gráfico 6 Logo de la versión 1.0 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado junto al primer dispositivo, el HTC Dream G1, el 23 de septiembre de 2008.
 - Integración con Google Service.
 - Web Browser con zoom, compatible con páginas web HTML y XHTML y multi-páginas mostradas como ventanas.
 - Android Market para descarga de aplicaciones y actualizaciones.
-
- **Android 1.1**
 - Liberado el 9 de febrero del 2009.
 - Actualización solo disponible para T-Mobile G1.

- **Android 1.5 (Cupcake)**

Gráfico 7 Logo de la versión 1.5 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 30 de abril de 2009.
- Linux Kernel 2.6.27.
- Posibilidad de grabar y reproducir videos a través del modo camcorde.
- Capacidad de subir videos a Youtube e imágenes a Picasa desde el dispositivo.
- Teclado con texto predictivo.
- Soporte para Bluetooth A2DP y AVRCP.
- Capacidad de conexión automática para conectar a auricular Bluetooth a cierta distancia.
- Nuevos widgets y carpetas que se pueden colocar en las pantallas de inicio.
- Transiciones animadas de pantalla.

- **Android 1.6 (Donut)**

Gráfico 8 Logo de la versión 1.6 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 15 de septiembre de 2009.
- Linux Kernel 2.6.29.
- Mejoras en Android Market.
- Interface integrada de cámara, filmadora y galerías.
- La galería permite la eliminación de varias fotos a la vez.
- Búsqueda por voz actualizada, con respuesta más rápida y mayor integración con aplicaciones nativas, incluyendo la posibilidad de marcar a contactos.
- Experiencia de búsqueda mejorada que permite buscar marcadores, historiales, contactos y páginas web desde la pantalla de inicio.
- Indicador de Batería.
- Actualización de soporte para CDMA/EVDO, 802.1x, VPN y text-to-speech.

- Soporte para resoluciones de pantalla WVGA.
- Framework de gestos y herramienta de desarrollo GestureBuilder.
- **Android 2.0 y Android 2.1 (Eclair)**

Gráfico 9 Logo de la versión 2.1 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 26 de octubre de 2009.
- Linux Kernel 2.6.29.
- Velocidad de hardware optimizada.
- Soporte para más tamaños de pantalla y resoluciones junto con una interfaz de usuario renovada.
- Soporta HTML5.
- Mejoras en lista de contactos, en el calendario y Google Maps v3.1.2.
- Soporte para Microsoft Exchange y múltiples cuentas de correo.
- Soporte para flash en la cámara.
- Zoom digital.
- MotionEvent mejorado para captura de eventos multi-touch.

- Bluetooth 2.1.
 - Fondos de pantalla animados.
 - Junto a esta versión fue liberado el SDK 2.0.1 y el SDK 2.1.
- **Android 2.2 (Froyo)**

Gráfico 10 Logo de la versión 2.2 de Android

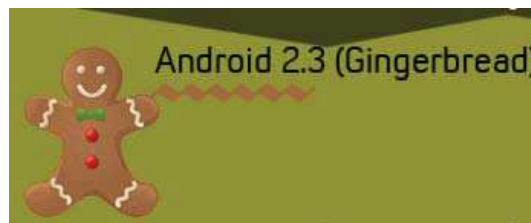


Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 29 de mayo de 2010.
- Linux Kernel 2.6.32.
- Optimización general del sistema Android, la memoria y el rendimiento.
- Implementación de JIT lo cual mejora la velocidad de las aplicaciones.
- Integra motor JavaScript V8 de Google Chrome en el web browser.
- Mejoras en Microsoft Exchange.
- Lanzador de aplicaciones mejorado con accesos directos a las aplicaciones de teléfono y Browser.
- Funcionalidad de Wi-Fi hotspot y tethering por USB.

- Actualización de Android Market.
 - Actualización del teclado a múltiples idiomas y su diccionario predictivo.
 - Marcación por voz y compartición de contactos por Bluetooth.
 - Soporta la instalación de aplicaciones en la memoria expandible.
 - Soporta Adobe Flash 10.1.
 - Mejora en resolución de la pantalla 4" 720px.
- **Android 2.3 (Gingerbread)**

Gráfico 11 Logo de la versión 2.3 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 6 de diciembre de 2010.
- Linux Kernel 2.6.35.7.
- Soporte para pantallas extra grandes y resoluciones WXGA y mayores.
- Soporte nativo para telefonía VoIP SIP.
- Soporte para reproducción de videos WebM/VP8 y decodificación de audio AAC.

- Nuevos efectos de audio como reverberación, ecualización, virtualización de los auriculares y refuerzo de graves.
- Funcionalidades de cortar, copiar y pegar disponibles a lo largo del sistema.
- Teclado multi-táctil rediseñado.
- Mejoras en la entrada de datos, audio y gráficos para desarrolladores de juegos.
- Recolección de elementos concurrentes para un mayor rendimiento.
- Soporte nativo para más sensores (como giroscopios y barómetros).
- Administrador de descargas para archivos grandes.
- Incluye un administrador de tareas.
- Cambio de sistema de archivos de YAFFS a ext4.
- Soporte nativo para múltiples cámaras.

- **Android 3.0, 3.1 y 3.2 (Honeycomb)**

Gráfico 12 Logo de la versión 3.0, 3.1 y 3.2 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 10 de Mayo de 2011.
- Mejor soporte para tablets.
- Escritorio 3D con widgets rediseñados.
- Sistema multitarea mejorado.
- Mejoras en el navegador web predeterminado, entre lo que destaca la navegación por pestañas, auto rellenó de formularios, sincronización de favoritos con Google Chrome y navegación privada.
- Soporte para video chat mediante Google Talk.
- Mejor soporte para redes Wi-Fi.
- Añade soporte para una gran variedad de periféricos y accesorios con conexión USB: teclados ratones, hubs, dispositivos de juego y cámaras digitales. Cuando un accesorio está conectado, el sistema busca la aplicación necesaria y ofrece su ejecución.
- Se añade soporte opcional para redimensionar correctamente las aplicaciones inicialmente creadas para móvil para que se vean bien en Tablets.

- **Android 4.0 (Ice Cream Sandwich)**

Gráfico 13 Logo de la versión 4.0 de Android



Fuente: <http://en.wikipedia.org/wiki>

- Liberado el 19 de Octubre de 2011.
- Versión que unifica el uso en cualquier dispositivo, tanto en teléfonos, tablets, televisiones, netbooks, etc.
- Interfaz limpia y moderna con una nueva fuente llamada "Roboto", muy al estilo de Honeycomb.
- Opción de utilizar los botones virtuales en la interfaz de usuario, en lugar de los botones táctiles capacitivos.
- Aceleración por hardware, lo que significa que la interfaz podrá ser manejada y dibujada por la GPU (Unidad de Procesamiento Gráfico), aumentando notablemente su rapidez, su respuesta y evidentemente, la experiencia de usuario.
- Multitarea mejorada. Añade la posibilidad de finalizar una tarea simplemente desplazándola fuera de la lista.
- Ha añadido un gestor del tráfico de datos de Internet. El entorno le permite establecer alertas cuando llega a una cierta cantidad de uso y desactiva el tráfico de datos cuando rebasa su límite.

- Los widgets están en una nueva pestaña, se despliegan en una lista similar a las aplicaciones en el menú principal.
- El corrector de texto ha sido rediseñado y mejorado, ofreciendo la opción de tocar en una palabra para que aparezca una lista con las diferentes opciones de edición y sugerencias de palabras similares.
- Se puede descartar las notificaciones que no son importantes y también desplegar la barra de notificaciones con el dispositivo bloqueado.
- La captura de pantalla, se la realiza pulsando el botón de bajar volumen y el botón de encendido.
- Nuevas utilidades en la aplicación de la cámara como la posibilidad de tomar fotografías panorámicas de forma automática.
- Android Beam es la nueva característica que permite compartir contenido entre teléfonos. Vía NFC (Near Field Communication), una nueva tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos a menos de 10cm
- Reconocimiento de voz y facial.
- Las carpetas son mucho más fáciles de crear, con un estilo de arrastrar y soltar.
- Un único y nuevo framework para las aplicaciones.
- Soporte nativo del contenedor MKV.

- Soporte nativo para el uso de Stylus (lápiz táctil). (Wikipedia - Historia de Actualizaciones, 2011)

Cada una de las versiones de Android desde su lanzamiento ha tenido una evolución acelerada acorde al desarrollo de la tecnología y de la demanda del mundo actual. Las mismas brindan a los usuarios y desarrolladores mejoras en el sistema con nuevas funcionalidades y herramientas.

2.6 DESARROLLO ANDROID

Una de las características importantes de Android es su código abierto y libre, con acceso total a los recursos del dispositivo, que permite la creación de aplicaciones innovadoras, ya que no está ligado a las restricciones generalmente asociadas con la propiedad de los frameworks de desarrollo móvil. Para el desarrollo en Android, Google liberó su SDK para compilar aplicaciones y poder emularlas en un equipo genérico. Esta emulación permite ejecutar el sistema operativo Android con todas sus características en la ventana de un computador con sistema operativo Linux, Windows o Mac; es como tener un dispositivo dentro de la ventana. Como un agregado se liberó también un plugin para el entorno de desarrollo Eclipse.

Las aplicaciones se escriben en lenguaje de programación Java, y se ejecutan en la máquina virtual Dalvik.

Android run time y Dalvik se encuentran sobre la capa del Kernel Linux, el cual se encarga de la interacción a bajo nivel de hardware, incluyendo los drivers y el manejo de la memoria. (Meier, 2009)

2.6.1 SDK de Android

El SDK de Android es una herramienta que incluye todo lo necesario para que los programadores puedan desarrollar aplicaciones para Android, emularlas, probarlas y depurarlas. Esta herramienta incluye:

- **API's de Android.-** son un conjunto de librerías núcleo del SDK que sirven para escribir las aplicaciones y que permiten el acceso de los programadores a la pila de Android (la arquitectura de Android). Estos API's son las mismas que utiliza Google para desarrollar las aplicaciones nativas de Android.
- **Herramientas de desarrollo.-** el SDK provee varias herramientas de desarrollo que permiten compilar el código y depurar las aplicaciones.
- **Emulador.-** como se indicó anteriormente, el emulador que incluye el SDK provee de un excelente ambiente de pruebas y puede emular cualquier versión de Android. Una de las limitaciones del emulador es el hardware ya que no se puede emular dispositivos como GPS o hacer llamadas.
- **Documentación.-** la documentación publicada en la web, foros, libros y en el propio SDK, contiene información sobre las versiones de Android, ejemplos de código, detalle de los componentes gráficos y ayuda que permite consultar cualquier duda de los desarrolladores y usuarios de Android.
- **Código de ejemplo.-** se incluye código seleccionado de ejemplo que indica algunas de las ventajas que ofrece el desarrollo para Android.
- **Soporte en línea.-** se lo encuentra en el sitio web oficial de Android <http://developer.android.com/index.html> en donde existen foros, videos, se comparte la experiencia en el desarrollo y ayuda a resolver inquietudes. Además existen muchos sitios en línea no oficiales que ayudan tanto a los desarrolladores expertos como principiantes. (Meier, 2009)

2.6.2 Arquitectura de las Aplicaciones

Las aplicaciones de Android esta compuestas por uno o más de los siguientes conceptos:

- **Conjunto de Vistas o Views.-** son los controles gráficos como grillas, botones, listas, web browser y mapas embebidos.
- **Proveedores de contenido.-** sirven para acceder los datos desde otra aplicación que lo solicite (como a los mapas, contactos, imágenes o GPS).
- **Administrador de recursos.-** proporciona acceso a todos los elementos que se incluyen directamente en el código como: imágenes, layouts, archivos de interfaz gráfica o archivos string (archivos con cadenas de texto que pueden estar en diferentes idiomas). Permite también gestionar estos elementos fuera del código de la aplicación.
- **Administrador de notificaciones.-** permite notificar al sistema operativo y por tanto al usuario los eventos que hayan ocurrido en una aplicación. Las notificaciones se puede visualizar como alertas en la barra de notificaciones de Android o como mensajes desplegados en la pantalla del dispositivo. Por ejemplo, los mensajes de texto que se reciben los dispositivos celulares y que se despliegan en la barra de notificaciones.
- **Administrador de actividades Android.-** es una de las piezas más importantes del sistema operativo. Se encarga de organizar y administrar la navegación del usuario por cada una de las aplicaciones que se encuentra en ejecución, permitiendole volver atrás cuando sea necesario. (Firtman, 2010)

2.6.3 Anatomía de una Aplicación

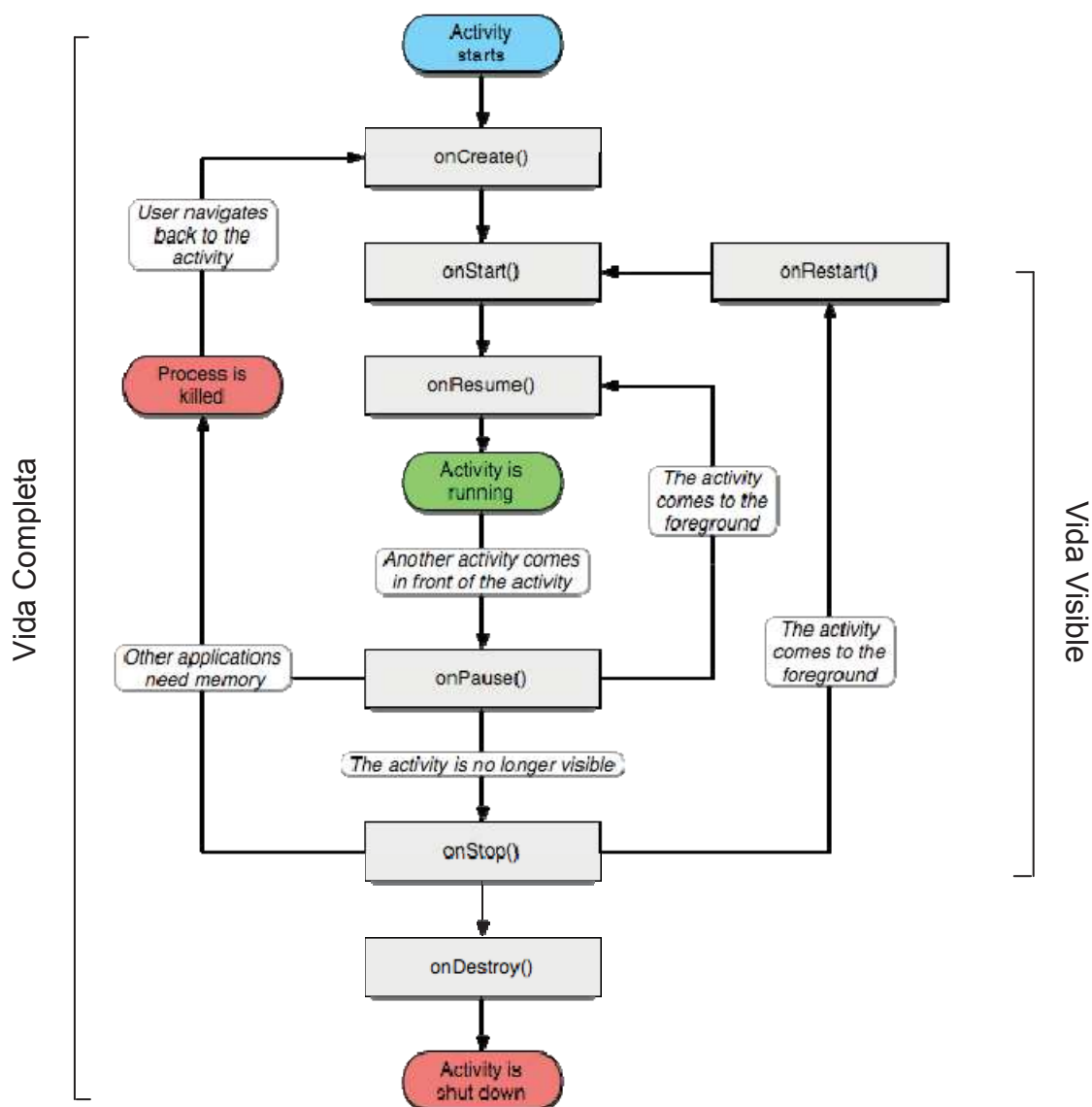
Para armar una aplicación se requieren de bloques de construcción lo cual va relacionado con la arquitectura de la aplicación. A continuación se detallan los bloques de construcción:

- **Actividades (Activities).**- es cada pantalla de la aplicación que se quiere mostrar. Algunas pueden no tener interface de usuario ya que pueden correr en background, es decir, en segundo plano.

Cuando una aplicación inicia, lo primero que aparece es la pantalla principal o Main Activity, cuando está ejecuta otra Actividad propia de la aplicación o externa, la anterior se pone en pausa y se envía a una pila. Esto es similar a cuando se navega por la web. Por otro lado, el administrador de actividades puede decidir eliminar una Actividad de la pila cuando lo crea conveniente.

La gráfico 14 muestra el ciclo de vida de un Actividad.

Gráfico 14 Ciclo de vida de un actividad (AndroidZona, 2011)



Elaborado por: El autor

En resumen, las actividades en su ciclo de vida tienen los siguientes estados:

- **Activo.**- en este estado la Actividad se está ejecutando y es visible en pantalla.
- **Pausada.**- en este estado la Actividad está todavía ejecutándose pero no tiene el foco, es decir, no es visible en pantalla.

- **Parada.-** en este estado la Actividad no es visible en pantalla y ya no se está ejecutando.
- **Intenciones.-** es una acción que se realiza desde el código. Las Intenciones permiten interconectar componentes de la misma o de distintas aplicaciones por medio del paso de mensajes. Por ejemplo, arrancar actividades o enviar eventos a varios destinatarios.
- **Receptor de Intenciones.-** se usa cuando se requiere que la aplicación reaccione ante algún evento externo como:
 - Llamada entrante.
 - Llegada de un SMS.
 - Detección de conexiones de red
 - Capturar intenciones de otras aplicaciones.

La aplicación puede estar cerrada, pero cuando la intención ocurre la aplicación se inicia automáticamente.

- **Servicios.-** es un código que se ejecuta en el sistema operativo sin ninguna interfaz gráfica (como una aplicación residente o un antivirus que se está ejecutando en segundo plano). Este servicio puede exponer funciones públicas para que otras aplicaciones las invoquen y también pueden invocar actividades si lo necesita. (Firtman, 2010)

Las aplicaciones Android generalmente se desarrollan con al menos uno de los cuatro bloques descritos anteriormente, ya que no todos son obligatorios. Una de las piezas importantes de las aplicaciones es el archivo llamado **AndroidManifest.xml**, al cual se le conoce como el Manifiesto y en donde se guardan los metadatos del proyecto Android, es decir, la información esencial y necesaria sobre el sistema operativo Android y sobre la aplicación.

2.6.4 Interfaz de Usuario

Las interfaces de usuario en Android funcionan basadas en una actividad, es decir, que una actividad contiene una interfaz de usuario.

A continuación se describen los componentes principales de una interfaz de usuario:

- **Vistas (Views).**- son clases heredadas de `android.view.View` y cada una representa un objeto rectangular en la pantalla, el cual maneja:
 - Medidas y disposición.
 - Entrada y salida de foco.
 - Desplazamiento entre objetos (o scrolling).
 - Gestos y expresión de teclas.

- **Widgets.**- son clases que hereda de `View` y que ya implementa una interacción en pantalla, es decir, es un control real. Por ejemplo: `Text`, `EditTex`, `Button`, `RadioButton`, `CheckBox` y `ScrollView`.

- **ViewGroups.**- son contenedores que permiten agrupar Vistas o Widgets en una sola zona. Dentro de estos agrupadores los más conocidos son los `Layouts` que ayudan a disponer los elementos en pantallas, entre otras funciones.

- **Layouts.**- permite agrupar u organizar las Vistas o Widgets de distintas maneras dependiendo el tipo de dispositivo o las resoluciones de pantalla, de esta manera los `Layouts` simplifican la reagrupación de los elementos en pantalla cuando se tiene diferentes resoluciones o cuando se cambia la orientación del dispositivo. Algunos de los más usados son:

- `FrameLayout`: pensado solo para ubicar un solo elemento.
 - `LinearLayout`: permite alinear el contenido horizontal o verticalmente.
 - `TableLayout`: permite definir filas y columnas.
 - `AdsoluteLayout`: permite definir cada elemento ubicándolos en x,y.
 - `RelativeLayout`: permite que cada elemento se ubique de manera relativa a otro.
- **Temas (Themes)**.- son la piel (o skin) que se aplican a los Widgets, es decir, que si se cambia de tema a los Widgets, estos automáticamente cambiarán su visualización. Android contiene Temas predefinidos y el desarrollador puede crear sus propios.

Es importante mencionar que las interfaces de usuario se pueden definir de manera declarativa a través de un XML (es lo recomendado) o de manera programática, es decir, a través de código Java. (Firtman, 2010)

3 CAPÍTULO III. HERRAMIENTAS DE DESARROLLO

En el desarrollo de este proyecto de tesis se utilizarán algunas herramientas que ayudarán en la codificación del aplicativo.

3.1 MOTODEV STUDIO FOR ANDROID

Es un IDE de desarrollo que ofrece la compañía Motorola, basado Eclipse y en el SDK de Google, diseñado especialmente para el desarrollo de aplicaciones Android.

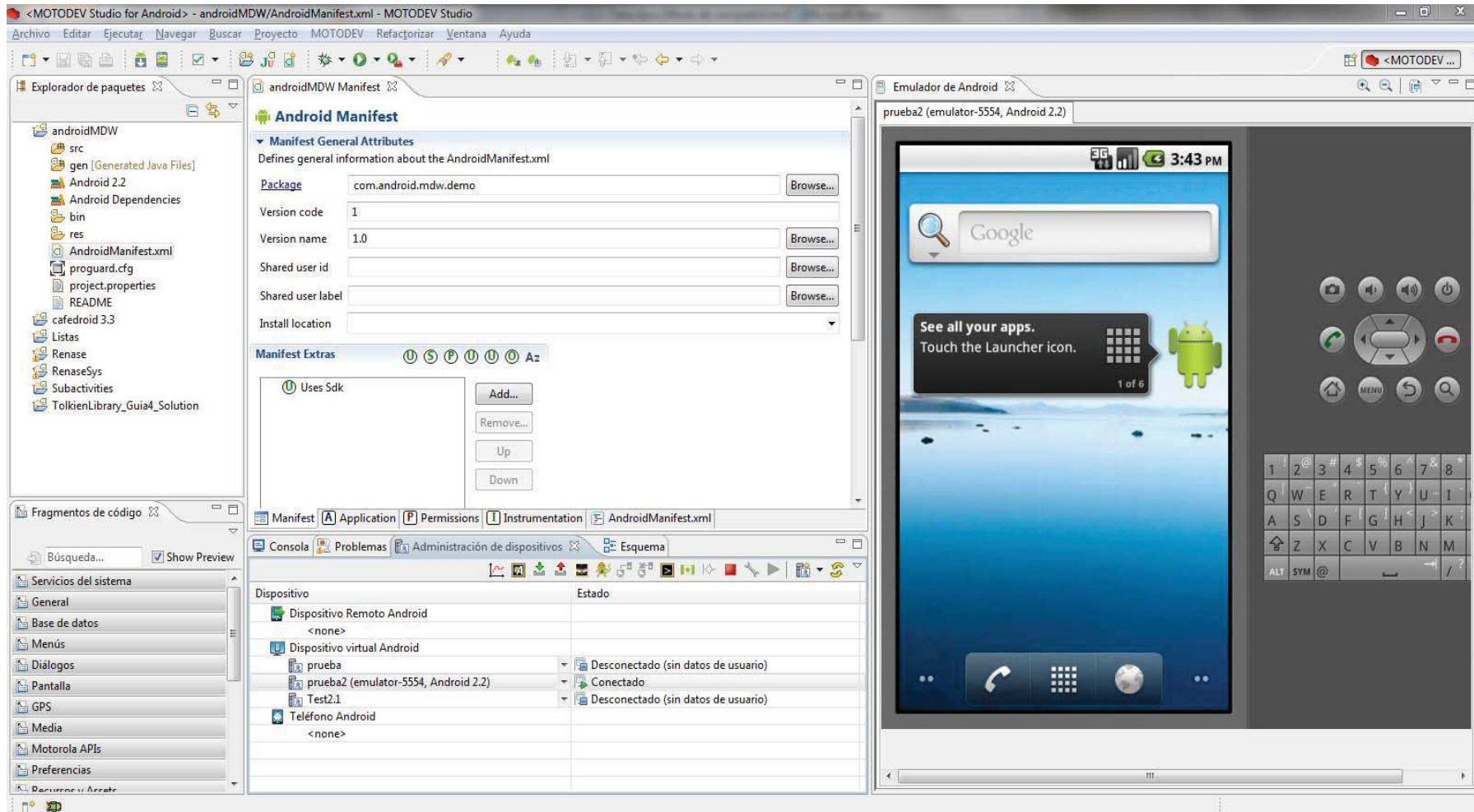
Motodev Studio incluye varias ventajas y mejoras para el desarrollo en Android que se describen a continuación:

- **Instalación.-** existen dos maneras sencillas de instalar Motodev Studio, desde un ejecutable que se lo descargará desde la página de Motorola <http://developer.motorola.com/> o se puede instalar como un plugin en el caso que ya se disponga de Eclipse.
- **Validación de Aplicaciones.-** esta característica detecta y repara condiciones inadecuadas de la aplicación, tales como permisos que se requieren o conflictos con las especificaciones del dispositivo.
- **Fragmentos de Código.-** son partes de código de uso frecuente, los cuales se pueden insertar en la aplicación para ayudar a resolver distintas tareas.
- **Gestor de Base de Datos.-** permite manipular y editar las bases de datos SQLite propias de Android.
- **Emulador Integrado.-** ejecuta el emulador en el mismo entorno de desarrollo sin necesidad de cambiar al emulador de terminal.

- **Administrador de Dispositivos.-** permite crear, ejecutar y manipular los dispositivos creados en el emulador; además, permite emular el envío de SMS y llamadas.
- **Asistente para la Creación de Aplicaciones.-** a través de este asistente se puede generar nuevos proyectos, clases necesarias para Android e instaladores con mucha facilidad.
- **Herramientas de Diagnostico.-** permite el uso de herramientas que detectan problemas de memoria en la aplicación antes de su distribución.

(GuajirAndro, 2012)

Gráfico 15 Entorno de desarrollo Motodev Studio for Android



Elaborado por: El autor

3.2 ADT PLUGIN DE ANDROID

El ADT (Android Development Tools) es un plugin diseñado para el IDE de desarrollo Eclipse, el cual crea un ambiente potente e integrado para el desarrollo de aplicaciones Android.

El ADT amplía las características de Eclipse para que pueda crear rápidamente proyectos de Android, diseñar interfaces de usuario, agregar los paquetes base de las APIs del framework de Android, utilizar las herramientas del SDK que ayudan a depurar y probar las aplicaciones, e incluso permite generar los ejecutables.apk de Android para la distribución de la aplicación. (Android Developers, 2012)

4 CAPÍTULO IV. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

En este capítulo se desarrollará el análisis y la planificación del sistema para la empresa Renase utilizando la metodología de desarrollo de software XP.

4.1 ANTECEDENTES

La empresa RENASE Cia. Ltda. es una empresa ecuatoriana que desde 1985, investiga, desarrolla, produce y comercializa, productos naturales de uso medicinal, alimenticio y cosmético, efectivos y seguros con los cuales contribuye al mejoramiento continuo de la calidad de vida de la sociedad.

La empresa vende sus productos a través de tiendas naturistas, por tanto su mercado objetivo son personas que consuman productos naturales para el cuidado de la salud.

La empresa elabora más de 30 productos medicinales, alimenticios y cosméticos, siendo su fuerte los productos de medicina natural.

4.2 IDENTIFICACIÓN DEL PROBLEMA

La empresa Renase actualmente presenta varias deficiencias en el proceso de pedidos desde la recepción de los mismos hasta entrega al cliente. Este proceso se lo hace manualmente sin el uso de ningún sistema. Al entrevistar al gerente general de la empresa se pudo detectar los siguientes problemas que se detallan a continuación:

1. Existen hasta 4 intermediarios en la solicitud del pedido. La primera persona en realizar el pedido es el cliente, quien hace saber al agente vendedor. Este a su vez se comunica con la empresa y transmite al asistente de ventas, quien finalmente pasa al jefe de bodega. El problema que se presenta en este proceso es que al pasar de persona a

persona, se corre el riesgo de una recepción equivocada y en varias ocasiones el pedido llega modificado al jefe de bodega.

2. La recepción de pedidos se transcribe en un cuaderno, en donde se coloca la fecha, el nombre del cliente, la ciudad, forma de pago, descripción del pedido y responsables del despacho, es decir, que se receptan escribiéndolos a mano. En este caso el asistente de ventas en varias ocasiones se equivoca en la escritura colocando mal la cantidad de unidades solicitadas o con una descripción del producto que crea confusión. Por ejemplo, existe tres presentaciones del producto Propóleo en Miel, cuando el asistente toma la orden escribe 10 propóleos, pero no se sabe de qué presentación es.

Los problemas no solo se presentan con la descripción de los productos, también con el cliente. Por ejemplo, al escribir los datos del cliente se olvidan de colocar el nombre completo de la persona o a su vez en que ciudad se encuentra. Existen clientes que tienen locales en varias ciudades, por lo tanto se debe colocar para que ciudad y local solicitan el pedido. Si se pierde esta información, la empresa tiene que intentar descifrar de quien se trata, lo cual es aún más complicado cuando existen homónimos en los nombres.

3. Se presentan errores en los detalles del pedido como la cantidad de productos o devoluciones. La empresa pierde tiempo cuando existen estos errores ya que necesita corregirlos y esto ocasiona que no exista una entrega a tiempo, es decir, que la empresa coloca el pedido en espera hasta solucionarlo y se toma de 5 a 8 días en despacharlo, lo cual provoca molestias para el cliente y baja su fidelidad para con la empresa.
4. Los agentes vendedores se demoran en entregar las solicitud de pedido, especialmente cuando se encuentran de gira. Esperan hasta el final de la gira para pasar todos los pedidos de los clientes que visitaron.

La empresa, por lo tanto, se ve en la necesidad de requerir un sistema que le permita minimizar los errores detallados. Los requerimientos para el sistema son:

- Reducción de intermediarios en la solicitud de pedido, es decir, que este pueda llegar a la empresa directa e inmediatamente.
- Una descripción y cantidad clara de los productos requeridos.
- Detalle de los clientes que permita identificarlos, saber en qué ciudad se encuentran y cuál es el trato que se le da.
- Que el agente de ventas o el cliente propiamente pueda hacer la solicitud desde el sitio de venta, sin necesidad de esperar al final del día para pasar el pedido o al final de la gira.

Luego de haber descrito los problemas que existen en la empresa y entender los requerimientos del sistema, se ha propuesto la creación de un sistema de pedidos móvil y que utilice la plataforma Android, tomando en cuenta que en la actualidad con el avance de la tecnología y la comunicación, los dispositivos móviles inteligentes con ésta plataforma se han difundido en el mercado ecuatoriano y son de fácil acceso para diferentes estratos de población. Este sistema aprovechará todas las ventajas que ofrece esta tecnología, como el acceso a Internet móvil, y el desarrollo y uso de aplicaciones que permiten integrar varios sistemas.

4.3 HISTORIAS DE USUARIO

Como se indicó en el capítulo I, la metodología de desarrollo XP a diferencia de las tradicionales utiliza las Historias de Usuario para especificar los requerimientos que en este caso RENASE como cliente tiene.

A continuación se detallan y analizan las Historias de Usuario conjuntamente con el cliente:

- **Historia de Usuario 1:**

Tabla 4 Historia de usuario 1

HISTORIA DE USUARIO No. 1					
Fecha	16/09/2011	Tipo de Actividad	<input checked="" type="checkbox"/> Nuevo	Prioridad Técnica	<input checked="" type="checkbox"/> Normal
			<input type="checkbox"/> Corrección		<input type="checkbox"/> Medio
			<input type="checkbox"/> Mejoras		<input type="checkbox"/> Alto
Referencia		Riesgo	<input checked="" type="checkbox"/> Ninguno	Prioridad Usuario	<input checked="" type="checkbox"/> Normal
			<input type="checkbox"/> Medio		<input type="checkbox"/> Medio
			<input type="checkbox"/> Alto		<input type="checkbox"/> Alto
Usuario		Cargo		Clasificación	Descripción de Productos
Descripción					
<ul style="list-style-type: none"> ✓ Lista de los productos de la empresa y su precio. ✓ Indicar el stock de los productos. ✓ Se debe indicar las características del producto. 					
Notas					
Seguimiento					
Estado		Comentario	Ultimo Seguimiento		
<input type="checkbox"/> Atendido					
<input checked="" type="checkbox"/> En Proceso					
<input type="checkbox"/> Finalizado					

Elaborado por: El autor

Análisis de la Historia de Usuario 1

Lista de Productos:

La aplicación debe ser capaz de desplegar en una ventana los productos de la empresa junto con su precio y de realizar una búsqueda rápida del producto ingresando el nombre.

Indicar el stock del producto:

El aplicativo debe indicar al usuario si existe el stock del producto requerido. En el caso de que exista el suficiente stock, el producto será entregado inmediatamente, de lo contrario el sistema deberá indicar que no existe el suficiente stock y que el producto será entregado en 4 días laborales. Solo se debe permitir una cantidad máxima de 100 unidades por producto.

Conocer las características de los productos:

Cada producto seleccionado debe indicar sus propiedades y presentación.

- **Historia de Usuario 2:**

Tabla 5 Historia de usuario 2

HISTORIA DE USUARIO No. 2					
Fecha	16/09/2011	Tipo de Actividad	<input checked="" type="checkbox"/> Nuevo	Prioridad Técnica	<input checked="" type="checkbox"/> Normal
			<input type="checkbox"/> Corrección		<input type="checkbox"/> Medio
			<input type="checkbox"/> Mejoras		<input type="checkbox"/> Alto
Referencia		Riesgo	<input checked="" type="checkbox"/> Ninguno	Prioridad Usuario	<input checked="" type="checkbox"/> Normal
			<input type="checkbox"/> Medio		<input type="checkbox"/> Medio
			<input type="checkbox"/> Alto		<input type="checkbox"/> Alto
Usuario		Cargo		Clasificación	Registro
Descripción					
✓ Registro de usuarios y actualización de datos. ✓ Para realizar un pedido es necesario ingresar a la cuenta con el nombre de usuario y clave.					
Notas					
Seguimiento					
Estado		Comentario	Ultimo Seguimiento		
<input type="checkbox"/> Atendido					
<input checked="" type="checkbox"/> En Proceso					
<input type="checkbox"/> Finalizado					

Elaborado por: El autor

Análisis de la Historia de Usuario 2

Creación de cuentas:

La aplicación debe tener la opción para que el cliente o agente vendedor pueda crear una cuenta. En la creación la aplicación debe solicitar los siguientes datos: nombre completo, cédula o RUC, dirección de correo electrónico, dirección del local o domicilio, ciudad, teléfono del local o domicilio, teléfono celular, nombre de usuario y clave.

Es importante indicar que la empresa validará y aprobará a cada cliente registrado mediante una confirmación telefónica o por correo electrónico. Si existe la confirmación el cliente podrá ingresar en el sistema y realizar pedidos, de lo contrario, el sistema bloqueará al cliente hasta recibir una confirmación.

Actualización o modificación de los datos del cliente:

El usuario debe tener la capacidad de poder modificar su información cuando lo desee.

Login o Sesión de Usuarios:

Para poder realizar el pedido, el usuario tiene que ingresar al sistema con su cuenta obligatoriamente; de lo contrario únicamente podrá ver los productos y su detalle. Si el usuario o clave son erróneos, se debe mostrar el mensaje “La clave o usuario son erróneos”. Si el usuario y clave son correctos, el sistema regresa a la pantalla principal pero con su sesión iniciada.

- **Historia de Usuario 3:**

Tabla 6 Historia de usuario 3

HISTORIA DE USUARIO No2					
Fecha	16/09/2011	Tipo de Actividad	<input checked="" type="checkbox"/> Nuevo <input type="checkbox"/> Corrección <input type="checkbox"/> Mejoras	Prioridad Técnica	<input checked="" type="checkbox"/> Normal <input type="checkbox"/> Medio <input type="checkbox"/> Alto
Referencia		Riesgo	<input checked="" type="checkbox"/> Ninguno <input type="checkbox"/> Medio <input type="checkbox"/> Alto	Prioridad Usuario	<input checked="" type="checkbox"/> Normal <input type="checkbox"/> Medio <input type="checkbox"/> Alto
Usuario		Cargo		Clasificación	Pedido
Descripción					
<p>✓ El pedido que realice el cliente será de los productos a elección del mismo, especificando las unidades que requiera. No se puede solicitar más de 100 unidades y puede escoger más de un producto.</p> <p>✓ Al finalizar el pedido, la aplicación muestra la lista de los productos escogidos por el cliente con su precio unitario y total. Además tiene la opción de eliminar cualquier producto o editar la cantidad y obligatoriamente el usuario debe ingresar la dirección de entrega.</p>					
Notas					
Seguimiento					
Estado		Comentario		Ultimo Seguimiento	
<input type="checkbox"/> Atendido					
<input checked="" type="checkbox"/> En Proceso					
<input type="checkbox"/> Finalizado					

Elaborado por: El autor

Análisis de la Historia de Usuario 3

Selección de Productos:

El usuario una vez que ha ingresado al sistema puede seleccionar uno o varios productos. Es obligatorio que ingrese la cantidad que solicita del producto al momento de seleccionarlo, siempre y cuando no exceda las 100 unidades.

Pantalla de Pedido:

El aplicativo indicará los datos del usuario que realiza el pedido, listará todos los productos seleccionados, la cantidad, precio y el total. El usuario puede modificar la cantidad de los productos seleccionados y eliminar cualquier producto de la lista. Mínimo podrá realizar el pedido con un producto.

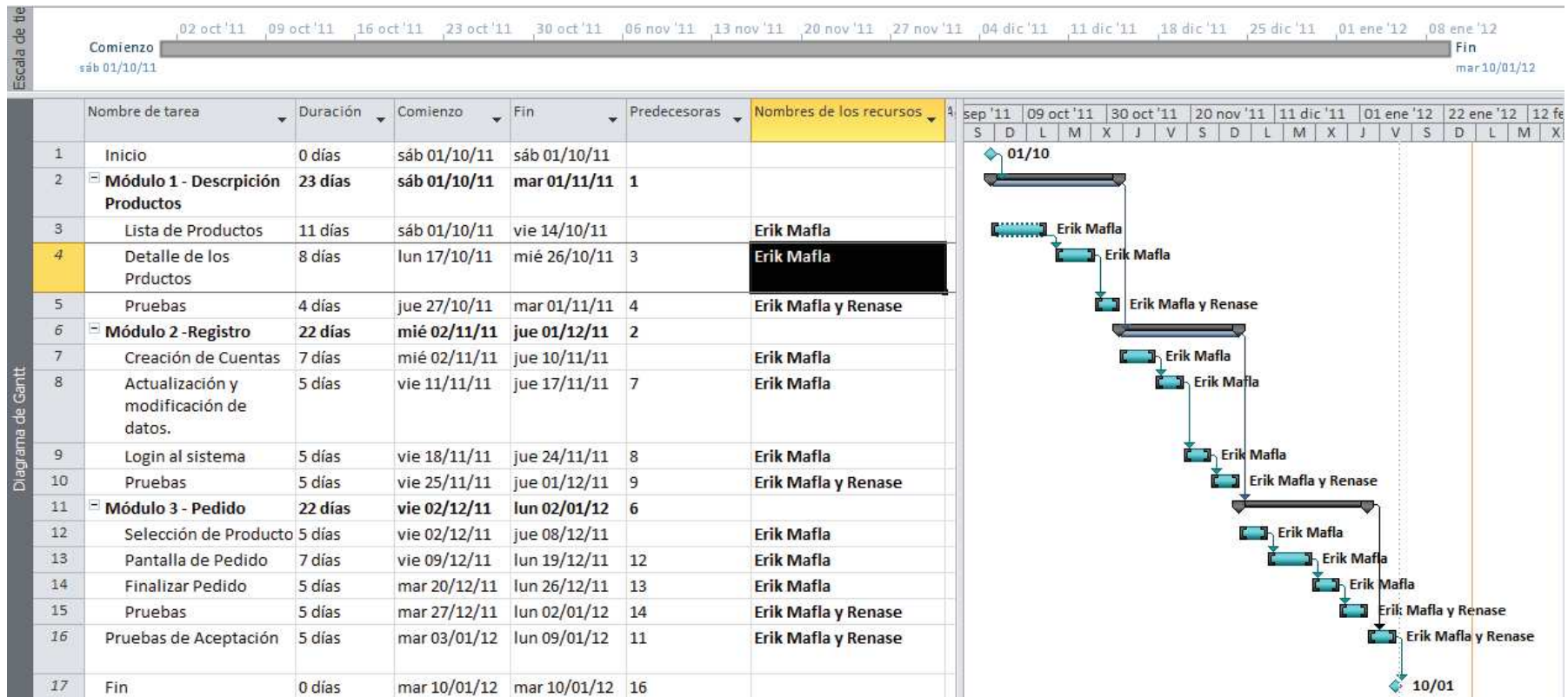
Finalizar Pedido:

El usuario confirmará el pedido o lo cancelará. Al finalizar, el pedido deberá ser guardado en la base de datos de la empresa para que ésta lo tome a través de su sistema interno.

4.4 PLAN DE ENTREGAS

En esta fase del proyecto se ha tenido una reunión con el cliente, en la cual se ha expuesto y aprobado el plan de entregas de la gráfico 16.

Figura 4.1 Plan de entregas del proyecto



Elaborado por: El autor

- **Plan de Entrega Etapa 1:**

Tabla 7 Plan de entrega etapa 1

FECHA:	Octubre 3 de 2011
EMPRESA:	RENASE CIA. LTDA.
HISTORIA DE USUARIO:	1
CLASIFICACIÓN:	Descripción de Productos

DETALLE: Como primera etapa se desarrollará el primer módulo Productos descrito en la Historia de Usuario 1.

NOTAS: Los tiempos determinados como entrega de cada actividad son acordados con el cliente y deberá ser respetado por las partes.

ACTIVIDADES	FECHA DE ENTREGA	COMENTARIOS
1. Lista de Productos. 2. Detalle del precio de cada producto. 3. Descripción del producto, indicando su stock, propiedades y presentación. 4. Pruebas finales del sistema.	1 mes	Inicia el 3 de Octubre de 2011. El cliente seguirá y realizará pruebas necesarias en el transcurso del desarrollo y si es necesario solicitará cambios sobre el módulo que se esta desarrollando en el momento posterior al tiempo programado de entrega. Cualquier cambio se cargará como un costo adicional.

Elaborado por: El autor

- **Plan de Entrega Etapa 2:**

Tabla 8 Plan de entrega etapa 2

FECHA:	Noviembre 3 de 2011
EMPRESA:	RENASE CIA. LTDA.
HISTORIA DE USUARIO:	2
CLASIFICACIÓN:	Registro

DETALLE: Como segunda etapa se desarrollará el módulo Registro descrito en la Historia de Usuario 2.

NOTAS: Los tiempos determinados como entrega de cada actividad son acordados con el cliente y deberá ser respetado por las partes.

ACTIVIDADES	FECHA DE ENTREGA	COMENTARIOS
1. Creación de cuentas de usuario. 2. Actualización o modificación de los datos del cliente. 3. Login o Sesión de Usuarios.	1 mes	Inicia el 3 de Noviembre de 2011. El cliente seguirá y realizará pruebas necesarias en el transcurso del desarrollo y si es necesario solicitará cambios sobre el módulo que se esta desarrollando en el momento, posterior al tiempo programado de entrega, cualquier cambio se cargará como un costo adicional.

Elaborado por: El autor

- **Plan de Entrega Etapa 3:**

Tabla 9 Plan de entrega etapa 3

FECHA:	Diciembre 3 de 2011
EMPRESA:	RENASE CIA. LTDA.
HISTORIA DE	
USUARIO:	3
CLASIFICACIÓN:	Pedido

DETALLE: Como tercera y última etapa se desarrollará el módulo Pedido descrito en la Historia de Usuario 3.

NOTAS: Los tiempos determinados como entrega de cada actividad son acordados con el cliente y deberá ser respetado por las partes.

ACTIVIDADES	FECHA DE ENTREGA	COMENTARIOS
1. Selección de productos. 2. Pantalla de pedido. 3. Finalizar pedido. 4. Pruebas.	1 mes	Inicia el 3 de Diciembre de 2011. El cliente seguirá y realizará pruebas necesarias en el transcurso del desarrollo y si es necesario solicitará cambios sobre el módulo que se esta desarrollando en el momento, posterior al tiempo programado de entrega. Cualquier cambio se cargará como un costo adicional. Se realizarán las pruebas respectivas de todo el sistema, a fin de probar el funcionamiento completo y que cumpla con lo requerido por el usuario.

Elaborado por: El autor

4.5 ITERACIONES

En base a las historias de usuario se han definido las iteraciones mostradas en la Tabla 10:

Tabla 10 Plan de Iteraciones para el desarrollo del aplicativo

Historia de Usuario N°	Nombre del Proceso	Prioridad	Riesgo	Esfuerzo	Iteración N°
1	Listado del productos.	Alta	Ninguno	Medio	1
1	Detalle de producto.	Media	Ninguno	Medio	1
1	Manejo de Stock de productos.	Baja	Ninguno	Bajo	1
2	Creación de cuentas de usuarios para clientes nuevos.	Alta	Ninguno	Alto	2
2	Actualización de datos.	Media	Ninguno	Alto	2
2	Ingreso y salida de sesión.	Media	Ninguno	Alto	2
3	Ingreso del pedido.	Alta	Ninguno	Alto	3
3	Detalle del pedido.	Alta	Ninguno	Medio	3
3	Envío del pedido.	Alta	Media	Medio	3

Elaborado por: El autor

- **Descripción de la Iteración 1**

Tabla 11 Iteración 1

Iteración:	1		
Módulo:	Productos	Responsable:	Erik Mafla
Prioridad:	Media	Tiempo Máximo:	28 días
Descripción de Actividades:			
<ul style="list-style-type: none"> - Despliegue de la lista de productos. - Búsqueda automática al ingresar el nombre. - Selección del Producto. - Mostrar el detalle del producto. - Manejo de stock del producto 			
Observaciones: Revisar detalle de producto con el cliente.			

Elaborado por: El autor

- **Descripción de la Iteración 2**

Tabla 12 Iteración 2

Iteración:	2		
Módulo:	Registro	Responsable:	Erik Mafla
Prioridad:	Media	Tiempo Máximo:	27 días
Descripción de Actividades:			
<ul style="list-style-type: none"> - Creación de cuentas de usuario nuevas. - Actualización de datos. - Cambio de clave de cuentas de usuario. - Ingreso y salida de sesión. 			
Observaciones:			

Elaborado por: El autor

- **Descripción de la Iteración 3**

Tabla 13 Iteración 3.

Iteración:	3		
Módulo:	Pedido	Responsable:	Erik Mafla
Prioridad:	Alta	Tiempo Máximo:	25 días
Descripción de Actividades:			
<ul style="list-style-type: none"> - Ingreso de pedido en el carrito de compras. - Despliegue del detalle del pedido en pantalla. - Edición del pedido. - Envío del pedido. 			
Observaciones:			

Elaborado por: El autor

4.6 CREACIÓN DE ESCENARIOS Y DE TARJETAS CRC

Es preciso conocer los elementos que intervienen en las tarjetas CRC para poder definirlos. Estos elementos son:

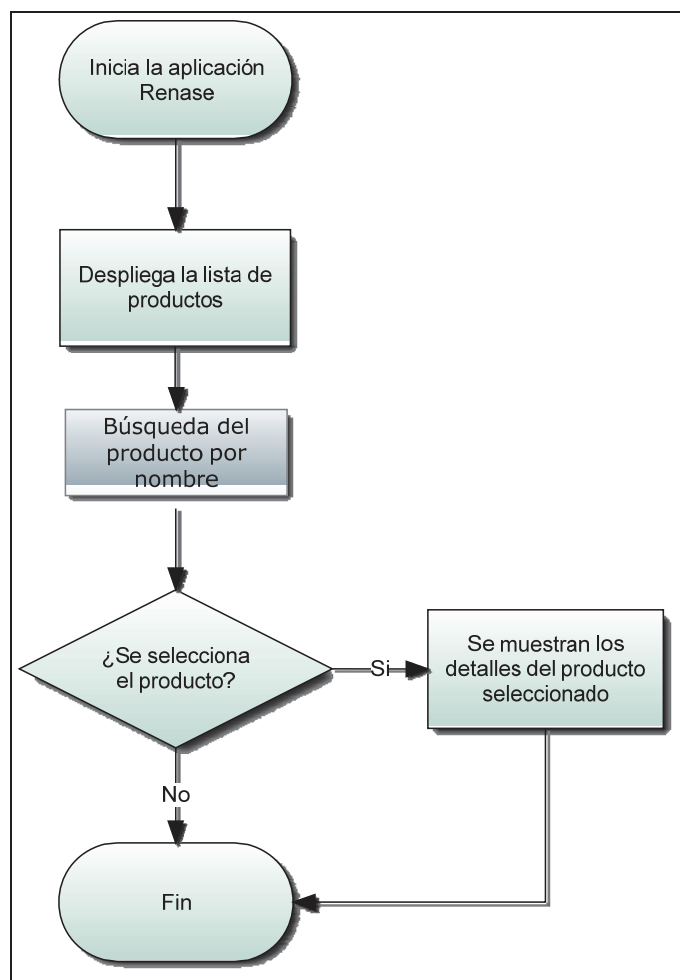
- Producto.
- Carrito de Compras (Pedido).
- Cliente.

Escenarios:

Es importante para la elaboración de las tarjetas CRC, detallar los posibles escenarios en donde se relacionan los elementos definidos en la aplicación.

Escenario 1: Búsqueda de Productos.

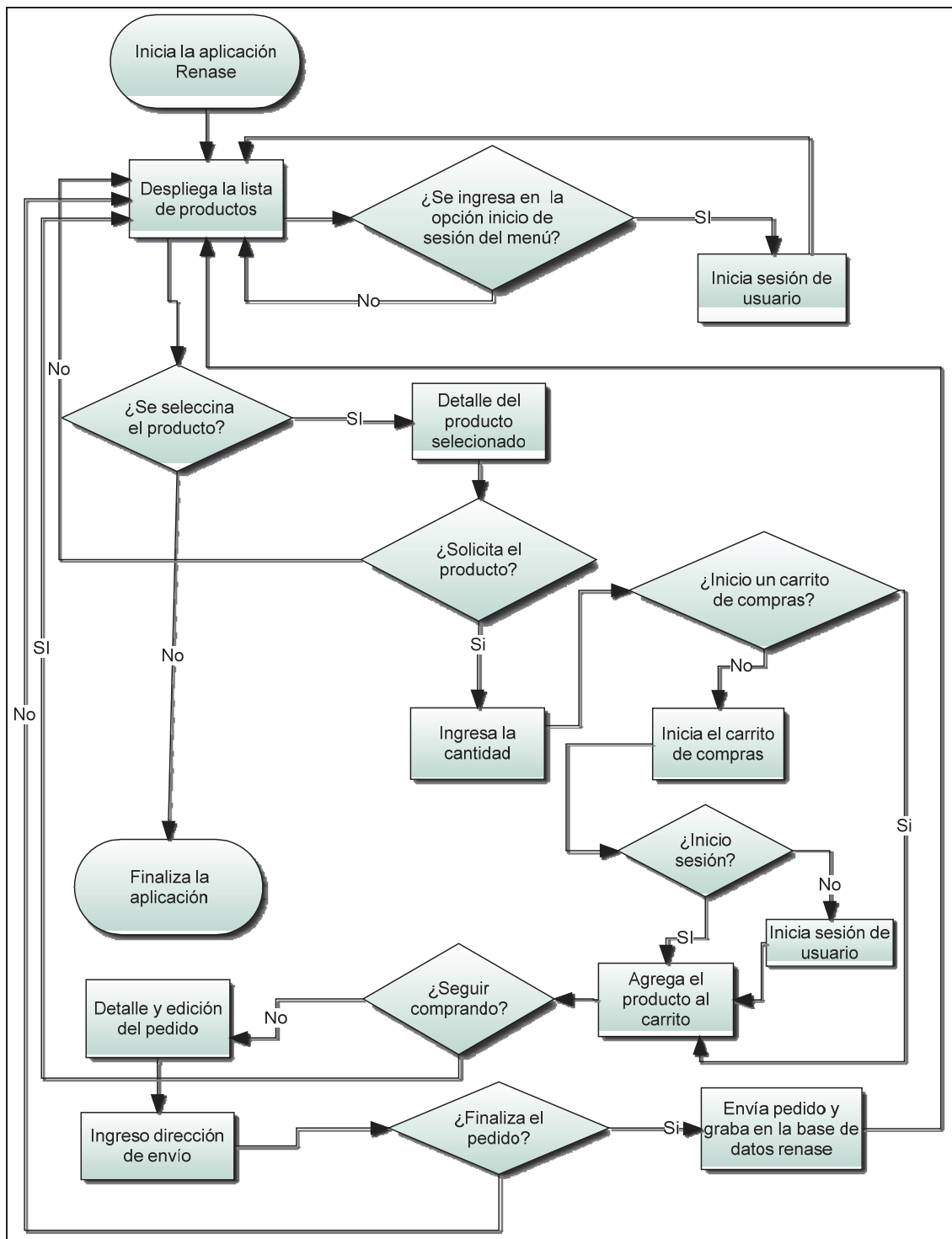
Gráfico 17 Búsqueda y despliegue de productos



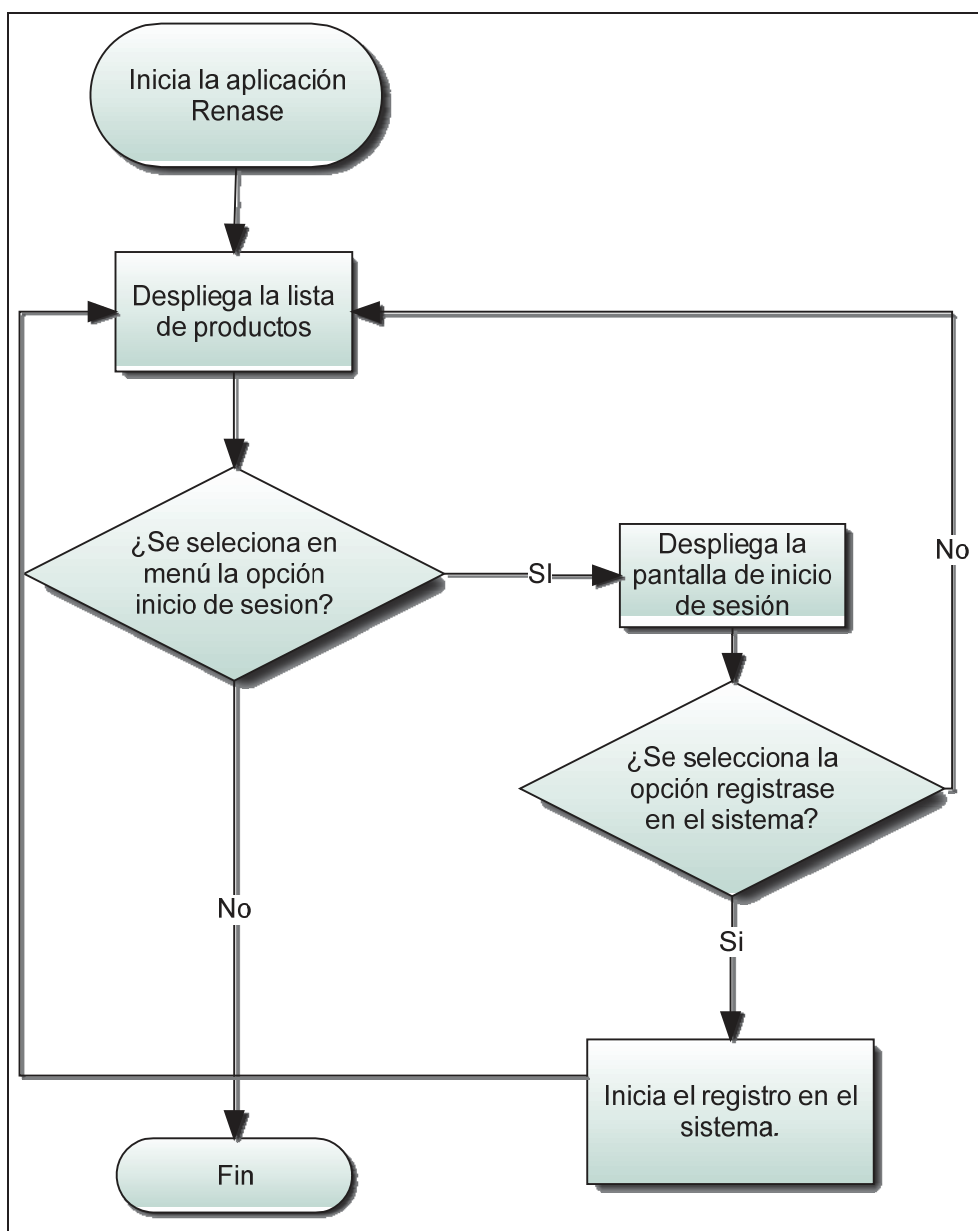
Elaborado por: El autor

Escenario 2: Realizar pedidos de productos.

Gráfico 18 Pedidos de productos



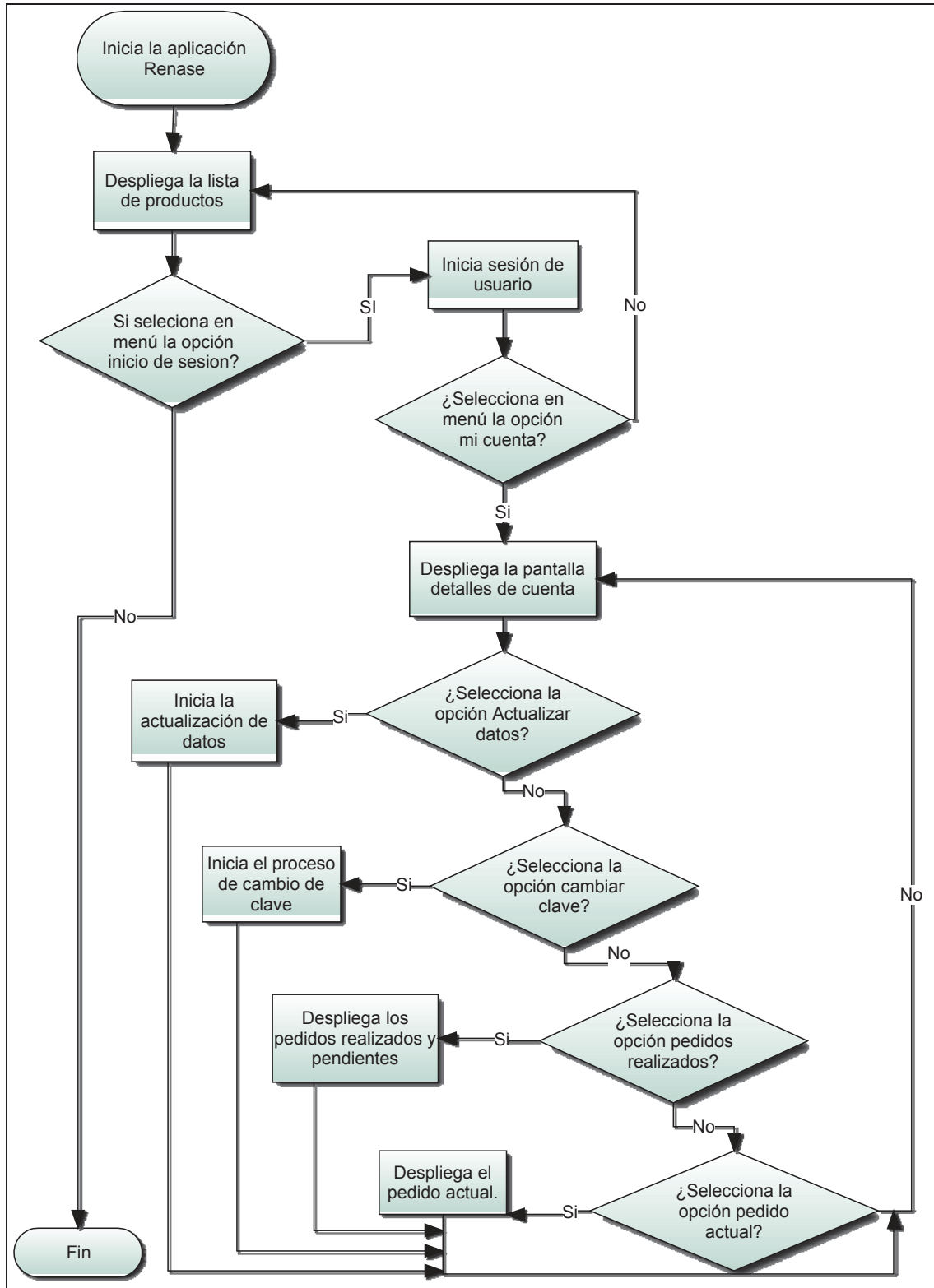
Elaborado por: El autor

Escenario 3: Registro de usuarios en el aplicativo.**Gráfico 19** Registro de usuarios en el aplicativo

Elaborado por: El autor

Escenario 4: Detalles de la cuenta de usuario.

Gráfico 20 Detalles de la cuenta de usuario



Elaborado por: El autor

A continuación se describen las tarjetas CRC que actúan como parte importante del diseño en la metodología de desarrollo XP.

Tabla 13 CRC Usuario.

Nombre de la clase: USUARIO	
Responsabilidad	Colaboración
Registrar un usuario.	Carrito de Compras
Actualizar la cuenta de usuario.	
Inicio de sesión.	
Cambio de clave.	

Elaborado por: El autor

Tabla 14 CRC Productos.

Nombre de la clase: PRODUCTO	
Responsabilidad	Colaboración
Desplegar lista de productos.	Carrito de Compras
Detalle del productos.	

Elaborado por: El autor

Tabla 15 CRC Carrito de Compras.

Nombre de la clase: CARRITO DE COMPRAS	
Responsabilidad	Colaboración
Ingresar productos.	
Finalizar pedido	
Edición de detalle pedido	

Elaborado por: El autor

4.7 DEFINICIÓN DE LA METÁFORA DEL SISTEMA

La metáfora del sistema ayuda en el desarrollo de la aplicación a llevar un orden lógico en la definición de clases y variables. A continuación se define la metáfora de los objetos, componentes, variables, métodos y herramientas que se usará en el desarrollo del sistema.

4.7.1 Definición de la Metáfora en los Objetos

Tabla 16 Definición de la metáfora en objetos

Objeto	Mayúsculas	Descripción	Ejemplo
Clases Java	Si	El nombre debe ser claro y especificar un objeto en singular.	Producto
Clase Activity	Si	El nombre debe ser claro y se especificará con un nombre claro el contenido general de la pantalla.	Login
Layout	No	El nombre debe ir relacionado con la clase activity que lo desplegará.	detalle_producto
Menú	No	El nombre debe ir relacionado con la clase activity después de la palabra 'menu'	MenuLogin

Elaborado por: El autor

4.7.2 Definición de la Metáfora en los Componentes

Tabla 17 Definición de la metáfora en componentes

Componente	Mayúsculas	Descripción	Ejemplo
LinearLayout	No	Sin detallar nombre	
TextView	Si	El nombre comenzará con las siglas tv, posteriormente se detallará un identificador de su función y luego la pantalla en la que se desplegará.	tvNombreUsuarioLogin
ListView	Si	El nombre comenzará con las siglas lv, posteriormente se detallará un identificador de su función y luego la pantalla en la que se desplegará.	lvListaProductosDetalleLista
EditText	Si	El nombre comenzará con las siglas et, posteriormente se detallará un identificador de su función y luego la pantalla en la que se desplegará.	etNombreActualizacionCliente
Button	Si	El nombre comenzará con las siglas btn, posteriormente se detallará un identificador de su función y luego la pantalla en la que se desplegará.	btnIngresoPedidoDetallePedido
ImageView	Si	El nombre comenzará con las siglas iv, posteriormente se detallará un identificador de su función y luego la pantalla en la que se desplegará.	ivLogoEmpresalInicio
Item	Si	El nombre comenzará con las siglas it, posteriormente se detallará un identificador de su función.	itLogin

Elaborado por: El autor

4.7.3 Definición de la Metáfora en las Variables y Métodos

Todos los tipos variables y métodos de cada actividad o clase se definirán con un nombre que detalle su función y especificando su control de acceso y retorno según sea el caso.

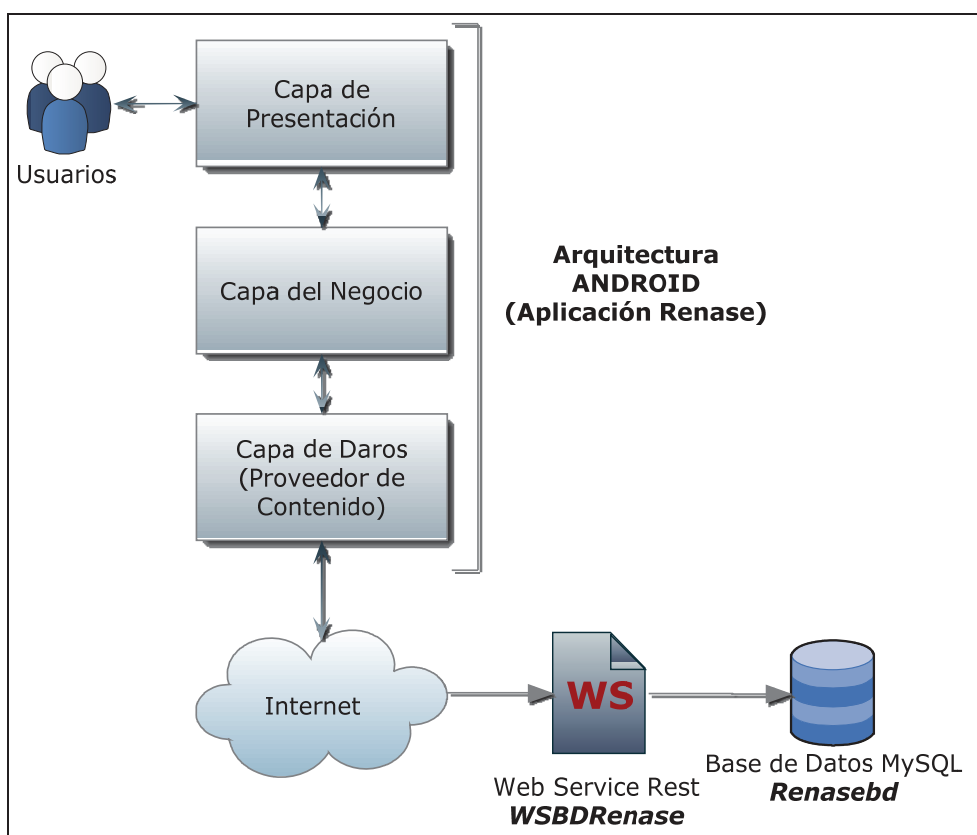
Ejemplo: `public String compraActual;`

4.8 DISEÑO DE LA APLICACIÓN

4.8.1 Arquitectura de la Aplicación

La aplicación Renase es similar a las aplicaciones Cliente-Servidor de tres capas como muestra en la gráfico 21:

Gráfico 21 Arquitectura de la aplicación para Renase



Elaborado por: El autor

La aplicación Renase se desarrollarán sobre la versión 2.2 de Android; de esta manera soportará cualquier versión superior, excluyendo las versiones anteriores. Al iniciarse la aplicación esta se debe conectar automáticamente con el web service WSBDRenase el cual se encargara de obtener y guardar la información de la base de datos MySQL renasedb.

A continuación se detalla cada capa de la arquitectura.

- La **capa de presentación** se compondrá por interfaces sencillas, intuitivas y amigables con el usuario, que permitan cumplir con la función del aplicativo.
- En la **capa de negocio** se encontrará las reglas del negocio, validaciones, cálculos y flujos que necesitará el aplicativo para su correcto funcionamiento. Por ejemplo, la cantidad mínima y máxima que se puede solicitar de un producto.
- La **capa de datos o proveedor de contenido** permite al aplicativo conectarse con la base de datos externa renasedb a través del web service WSBDRenase, el cual a se ha desarrollado utilizando la arquitectura RestFul y se encarga de obtener, insertar, actualizar o eliminar la información que requiera o envíe el aplicativo.

En la tabla 18 se muestra los métodos del web service WSBDRenase.

Tabla 18 Métodos del web service WSBDRenase.

Nombre Método	Acción	Detalle
listaProductos	Get	Obtiene la lista de productos de la empresa.
listaClientes	Get	Obtiene la lista de clientes de la empresa
insertarCliente	Post	Inserta el registro de un nuevo cliente.
actualizarCliente	Post	Actualiza la información del cliente.
insertarPedido	Post	Inserta el pedido.
insertarProductosPedido	Post	Inserta los productos del pedido.
obtenerPedidos	Get	Obtiene el pedido.
obtenerDetallePedidos	Get	Obtiene el detalle del pedido.
listProductosPedido	Get	Obtiene la lista de productos del pedido.

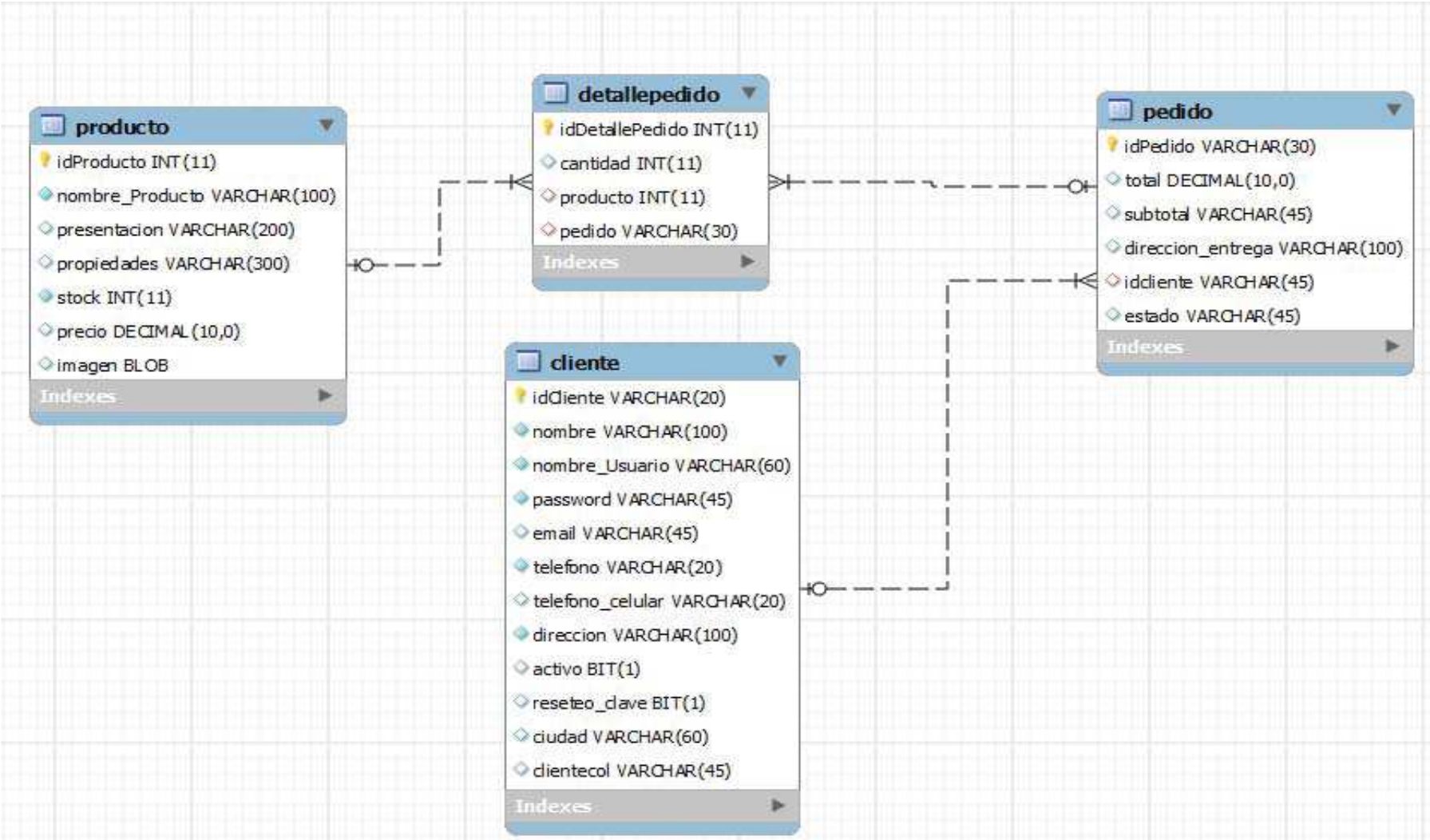
Elaborado por: El autor

La base de datos MySql renasedb y el web service WSBDRenase, que corre en un servidor de aplicaciones GlassFish Server 3.1.1, se instalaran en un equipo servidor dedicado para este uso en la empresa y se configurará el router de la empresa con un port forwarding para que a través de su IP pública y el puerto se conecten al web service WSBDRenase.

4.8.2 Diseño de la Base de Datos

En la grafica 4.7 se muestra el diseño de la base de datos **renasedb**, la cual fue desarrollada en base al análisis realizado a las historias de usuario.

Gráfico 22 Diseño de la base de datos renasedb



Elaborado por: El autor

5 CAPÍTULO V. IMPLEMENTACIÓN Y PRUEBAS

5.1 IMPLEMENTACIÓN

En esta fase del proyecto se ha tomado muy en cuenta las características de la metodología XP como son:

- **La comunicación con el cliente.-** se mantuvo una comunicación directa con el cliente y sus funcionarios en el desarrollo del sistema, quienes aportaron con sus comentarios y correcciones en la construcción del aplicativo, evitando así los cambios sobre los módulos una vez terminados.
- **La simplicidad al escribir el código.-** en este punto se tuvo algunos problemas con implementación de varias pantallas por la utilización de código excesivo, lo cual con la investigación y cambios constantes sobre estos se logró optimizar el funcionamiento y disminuir las líneas de código siendo mucho más fácil su implementación y se logró posteriormente un desarrollo más eficaz.
- **La refabricación o reutilización del código.-** fue uno de los aspectos más importantes en el desarrollo del sistema, logrando acelerar la codificación a medida que el proyecto avanzaba.
- **Pruebas unitarias.-** las pruebas unitarias ayudaron a corregir varios errores en la implementación de cada módulo, evitando así que en la integración se tuviera errores.

Nota: Como este proyecto fue desarrollado por una sola persona la Programación en Pares no se utilizó.

5.2 PRUEBAS DE UNIDAD Y ACEPTACIÓN

Las pruebas de unidad fueron realizadas en el desarrollo, probando cada módulo o fragmento de código antes de ser liberado o publicado. Las pruebas de aceptación se realizaron conjuntamente con el cliente para verificar cada historia de usuario.

Las tablas 19, 20 y 21 muestran los resultados de las pruebas de aceptación o de caja negra realizados.

Tabla 19 Prueba de caja negra - Descripción de productos.

Nombre de la clase	Nombre Prueba Caja Negra	Descripción	Resultado
Producto	Despliegue	Verificar que los productos se desplieguen y muestre la información requerida al iniciar la aplicación.	ok
	Búsqueda	Verificar que se pueda realizar la búsqueda por texto de los productos sobre la lista desplegada en un inicio.	ok
	Detalle	Verificar que al presionar sobre un producto en la lista se despliegue la información del producto requerido.	ok

Elaborado por: El autor

Tabla 20 Prueba de caja negra – Registro

Nombre de la clase	Nombre Prueba Caja Negra	Descripción	Resultado
CLIENTE	Inicio de Sesión (Login)	Verificar que un cliente ya registrado ingrese al sistema usando sus credenciales.	ok
	Mi Cuenta	Verificar que una vez que el cliente se encuentre registrado en el sistema pueda acceder a su cuenta a través del menú “Mi cuenta”	ok
	Actualización de Datos	Verificar que el cliente pueda actualizar su información y que el aplicativo valide los campos obligatorios.	ok
	Cambio de Clave	Verificar que el cliente pueda cambiar su clave.	ok
	Pedidos Realizados	Verificar que el cliente pueda revisar sus pedidos pendientes en el aplicativo y sus detalles.	ok
	Pedido Actual	Verificar que el cliente pueda desplegar el pedido actual y si no lo tiene validar mensaje.	ok
	Registro Nuevo Cliente	Verificar que el sistema permita registrar nuevos usuarios.	ok
	Despliegue de Mensajes	Verificar mensajes de registro, login y usuario.	ok

Elaborado por: El autor

Tabla 21 Prueba de caja negra - Pedido

Nombre de la clase	Nombre Prueba Caja Negra	Descripción	Resultado
Pedido	Ingreso de cantidad	Verificar que restricciones y mensajes e ingreso de cantidad antes de comprar un producto.	ok
	Compra o Agregar Producto	Verificar que una vez ingresada la cantidad se agreguen los productos al carrito de compras y verificar el mensaje posterior.	ok
	Despliegue del Pedido	Verificar que el pedido se despliegue con toda su información y se puedan editar los productos elegidos o eliminarlos.	ok
	Envió del Pedido	Verificar que el envío se haya realizado correctamente y que se guarde en la base de datos	ok
	Cancelar Pedido	Verificar que el pedido sea cancelado y el despliegue de los mensajes de alerta.	ok
	Restricciones	Verificar que para realizar un pedido el aplicativo indique que es necesario registrarse en el sistema.	ok

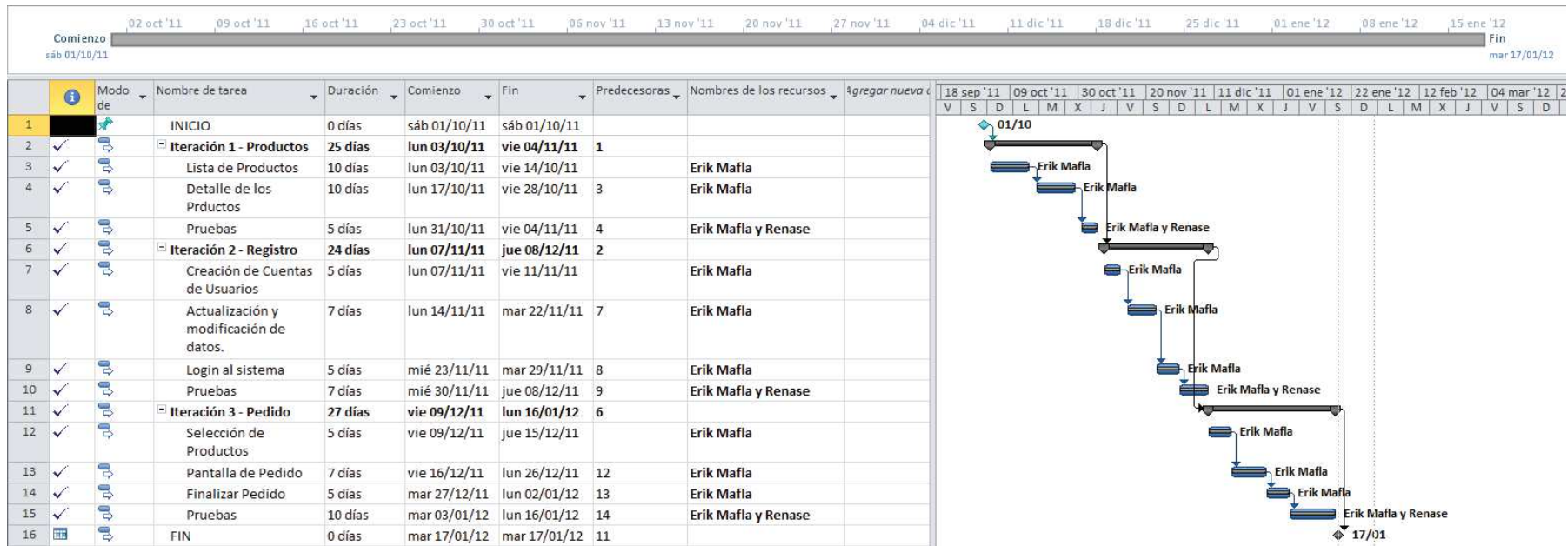
Elaborado por: El autor

Siendo estas pruebas satisfactorias y no indicando ningún cambio por el cliente se considera que el desarrollo es correcto cumpliendo con los tiempos establecidos en el plan de entregas.

5.3 DESARROLLO DE ITERACIONES

En la grafica 5.1 se muestra los tiempos tomados en el desarrollo de cada iteración y por ende de la aplicación.

Gráfico 23 Desarrollo de iteraciones



Elaborado por: El autor

5.4 DOCUMENTACIÓN DE LA APLICACIÓN

Renase es una aplicación desarrollada exclusivamente para la empresa Renase Cia. Ltda. la cual permite realizar pedidos en línea desde dispositivos móviles con sistema operativo Android.

A continuación se describen los detalles técnicos de la aplicación.

5.4.1 Plataforma

La aplicación Renase se encuentra desarrollada en el IDE de desarrollo Motodev Studio for Android utilizando como lenguaje JAVA y el SDK de Android v.2.2.

La aplicación opera en dispositivos que cumplan con los siguientes requisitos de plataforma:

- **Sistema Operativo:**
 - Android v.2.2 o versiones superiores.

- **Memoria Ram:**
 - 192 mb o superior.

- **Procesador:**
 - Un solo nucleó con velocidad de CPU de 800MHz o superior.

- **Network:**
 - Conexión a datos GRS/EDGE.
 - Wi-Fi 802.11 b/g/n (Opcional).

- **Pantalla**

- La aplicación fue diseñada para dispositivos con pantalla de 3.7" WVGA o 4" WVGA. Otras medidas de pantalla pueden causar que los objetos no se muestren correctamente (Los dispositivos de la empresa tienen esta resolución de pantalla y por esta razón se la selecciono).

6 CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES

Después de haber concluido con la aplicación móvil para la empresa **Renase** utilizando la metodología **EXTREAM PROGRAMMING** se pudo llegar a las siguientes conclusiones y recomendaciones.

6.1 CONCLUSIONES

- La aplicación móvil desarrollada permite optimizar el proceso de recepción y despacho de pedidos de la empresa Renase, eliminando retardos y errores que dañaban la imagen corporativa de la misma. La aplicación permite el envío en línea de los pedidos a la base de datos MySQL, agilizando de esta manera el proceso de facturación y despacho. Además perfecciona la solicitud de pedidos, al ser un sistema gráfico y de fácil manejo, ya que despliega en la pantalla los productos de la empresa, permite realizar búsquedas, visualizar el stock, revisar el detalle de los mismos, controlar las cantidades solicitadas, verificar y modificar el pedido antes de enviarlo, eliminando así los errores que se presentaban comúnmente al tomarlos de manera escrita y que causaban confusiones, retardos en las entregas y en el peor de los casos el envío de las ordenes incompletas o erróneas. De esta manera se ha logrado la satisfacción del cliente y mejorado la imagen de la empresa.
- La creación de la aplicación móvil permitió a la empresa Renase llevar un catálogo en línea de todos sus productos. La aplicación obtiene los productos desde la base de datos MySQL y los despliega de manera gráfica, dando a conocer a los vendedores las principales características, presentación, precio, imagen y stock, convirtiéndose en una fuente de consulta que además se actualiza constantemente, evitando el uso de folletos o vademécums que tienden a desactualizarse.

- Se comprobó que la metodología de desarrollo ágil XP asegura la calidad del software durante todo su ciclo de vida y que sus valores son fortaleza de la misma. Durante el desarrollo de la aplicación se obtuvieron funcionales al término de cada iteración, lo cual representa el éxito en el desarrollo y el interés de la empresa Renase en continuarlo. El uso de sus valores como la comunicación, la simplicidad y la retroalimentación permitieron trabajar conjuntamente con la empresa ejecutando pruebas continuas de la aplicación con el usuario final, realizar un desarrollo más sencillo y flexible a cambios, del cual se obtuvo como resultado una aplicación óptima y conforme a las necesidades de Renase.
- Android ha demostrado ser un sistema operativo acorde a las necesidades tecnológicas actuales. Gracias a sus características como su arquitectura y código abierto, han permitido a fabricantes y usuarios modificarlo según sus necesidades y requerimientos; además permite a las aplicaciones creadas acceder a todos los recursos del dispositivo y del sistema. Android también ofrece ventajas como el ser un sistema multitarea, integrar SQLite que permite gestionar bases de datos relacionales, libertad a los propietarios de instalar lo que deseen en sus terminales y la capacidad de implantarse en una gran variedad de dispositivos como netbooks, microondas, lavadoras, relojes, navegadores GPS, etc. Estas características y ventajas han permitido a Android ser un sistema que se encuentra a la vanguardia tecnológica y que satisface las necesidades de cada tipo de usuario, logrando ser uno de los más vendidos actualmente.
- Android se acopló perfectamente a los requerimientos del proyecto, permitiendo el desarrollo de aplicativo sin presentar problemas o limitaciones. Las características de Android como el acceso a internet móvil y WI-FI permitieron al aplicativo el envío inmediato de los pedidos. El SDK proporciono herramientas como el emulador y el entorno gráfico de desarrollo, que facilitó realizar la codificación, diseño y pruebas,

proporcionando un ambiente adecuado para crear la aplicación y que además permitió conocer las bondades que ofrece este sistema.

- Las tecnologías móviles usadas hoy en día por las empresas en dispositivos como celulares inteligentes y tabletas son herramientas muy útiles para la gestión empresarial. Mejoran la comunicación interna, agilizan procesos, minimizan errores, permiten una mejor organización y en general satisfacen las necesidades que las empresas presentan a un bajo costo, gracias al uso y desarrollo de aplicaciones, que explotan al máximo las ventajas que ofrecen los dispositivos como el uso de internet móvil, WI-FI, GPS o cámaras digitales. El desarrollo de la aplicación para Renase es un claro ejemplo del impacto que esta tecnología causa, ya que con el uso de características como el internet móvil, le permitió a la empresa mejorar su proceso de pedidos e incluso en un futuro se podría capacitar directamente a sus principales distribuidores para que realicen éstos usando la aplicación, sin necesidad de la visita de un vendedor. Otras empresas utilizan la tecnología móvil para el cobro en línea e inalámbrico de los productos de un local, utilizando la cámara digital como un lector de código de barras y las redes inalámbricas Wi-Fi para conectar su aplicación con su sistema de facturación; de esta manera, se optimiza el cobro y causa una satisfacción del cliente al darle comodidad y ahorro de tiempo, evitando que se acerque a la caja y que haga largas filas para cancelar.

6.2 RECOMENDACIONES

Al finalizar éste proyecto me permito sugerir las siguientes recomendaciones.

- Se recomienda para el desarrollo de software, el uso de la metodología Extream Programmig, la misma que brinda eficiencia, orden, eficacia y seguridad en el proceso, evitando perdidas de dinero, tiempos y recursos.

- Se recomienda a Renase el mantenimiento mensual de la aplicación y sus bases de datos, de esta manera se evitará la desactualización de la información y evitara que el aplicativo tenga errores que podrían ocurrir en un futuro con las actualizaciones del sistema.
- Se recomienda aplicar a los negocios la tecnología móvil que puede llegar a ser un excelente aliado usándola correctamente.
- Se recomienda a Renase en un futuro subir el aplicativo a Google Play para de esta manera no solo pueda ser usado por sus agentes vendedores y clientes actuales sino por el mercado que ahora disfruta de esta tecnología.
- Se recomienda a la empresa realizar la validación de usuarios que se registran en el sistema mediante usando el registro en la base de datos y activándolo usando el campo activo.

REFERENCIAS

Libros:

- Bello, C. L. (2003). Una Breve Descripción de Java. Retrieved from Una Breve Descripción de Java
- Canós, José, Patricio Letelier y Ma Carmen Penadés. (2004). Metodologías Ágiles en el Desarrollo de Software. Valencia: Universidad Politécnica de Valencia.
- Cortizo, José, Diego Expósito Gil y Miguel Ruiz Leyva. (2011). eXtreme Programming.
- Firtman, M. (2010). Android Basico Desarrollo Para dispositivos Moviles. Buenos Aires: Video2brain.
- Holzner, S. (2005). Java 2. Anaya.
- Letelier, Patricio y Penadés, M^a Carmen. (2010). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia: Universidad Politécnica de Valencia.
- Meier, R. (2009). Professional Android Application Development.

Documentos de Internet:

- Android Developers. (2011). Android Developers. Retrieved Noviembre 2011, from <http://developer.android.com/guide/basics/what-is-android.html>
- Android Developers. (2012). Android Developers. Retrieved 2012, from <http://developer.android.com/sdk/eclipse-adt.html>
- Android.com. (2011). Android Developers. Retrieved Noviembre 2011, from <http://developer.android.com/guide/basics/what-is-android.html>
- Androideity. (2011). Androideity. Retrieved 2011, from <http://androideity.com/2011/07/04/arquitectura-de-android/>
- AndroidZona. (2011). AndroidZona. Retrieved 2011, from <http://www.androidzona.net/programacion-android-el-ciclo-de-vida/>

- Corporation, I. (2012). Corporation, Instartius. Retrieved 2012, from <http://instartius.com/blog/?p=289>
- Edgedg, b. (2010). buenastareas. Retrieved 2011, from <http://www.buenastareas.com/ensayos>
- Guajir, Andro. (2012). Instalación de MOTODEV Studio for Android. Retrieved 2012, from <http://guajirandro.blogspot.com/2012/03>
- Instartius Corporation. (2012). Instartius Corporation. Retrieved 2012, from <http://instartius.com/blog/?p=289>
- Jeffries, R. (2001, 11 8). XProgramming. Retrieved 2011, from <http://xprogramming.com/book/whatisxp/>
- Joskowicz, I. (2008). Reglas y Prácticas en eXtreme Programming. Retrieved from <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Mundo Manuales. (2011). Mundo Manuales. Retrieved 2011, from <http://www.mundomanuales.com/telefonos/telefonos-moviles/que-es-android-caracteristicas-y-aplicaciones-4110.html>
- Oracle Corporation. (2011). Java. Retrieved 2011, from <http://www.oracle.com/technetwork/java/javase/namechange>
- Oracle Corporation. (2011). Mysql Dev. Retrieved 2011, from <http://dev.mysql.com/doc/refman/5.0/es/features.html>
- Wells, D. (2000). Extream Programming. Retrieved from <http://www.extremeprogramming.org/map/iteration.html>
- Wikipedia - Historia de Actuaizaciones. (2011). Historia de Actuaizaciones. Retrieved 2011, from <http://es.wikipedia.org/wiki/Android>
- Wikipedia. (2011). Android version history. Retrieved Noviembre 2011, from http://en.wikipedia.org/wiki/Android_version_history, Android version history
- Wikipedia. (2011). Android. Retrieved Noviembre 2011, from <http://es.wikipedia.org/wiki/Android>
- Wikipedia. (2011). Java lenguaje de programación. Retrieved Diciembre 2011, from <http://es.wikipedia.org/wiki/Java>
- Wikipedia. (2011). Telefonos Inteligentes. Retrieved Noviembre 2011, from http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente

GLOSARIO

CLR (Common Language Runtime): Son instrucciones para una maquina virtual y son ejecutadas indirectamente a través del compilador Just In Time (JIT), permiten generar en varios lenguajes un programa es independiente de la plataforma.

Camcorde.- es un dispositivo electrónico portátil para grabar imágenes de video y audio en un dispositivo de almacenamiento.

Framework: es una estructura conceptual y tecnológica definida que puede ser usada por otro proyecto de software con el fin de facilitar la organización y el desarrollo.

Iteración: es la repetición de una serie de instrucciones en un programa de computadora.

JIT “Just in time”.- es una filosofía que define la forma en que debería optimizarse un sistema de producción.

Kernel: es un software que constituye la parte más importante del sistema operativo.

Linux: es un núcleo de sistema operativo libre basado en Unix.

Software Libre: es la denominación al software que puede ser usado, copiado, estudiado, modificado, y redistribuido libremente.

XP (Extreme Programming): Metodología de Programación Extrema creada por Kent Beck que se base en principios y prácticas.

ANEXOS

ANEXO 1

Manual de Usuario de la Aplicación Renase

A continuación se detalla el uso y del aplicativo Renase.

Ingreso al Aplicativo

Al iniciar el aplicativo se debe presionar sobre el logo de la empresa para ingresar.



A continuación automáticamente se desplegarán la pantalla donde se listan productos.

Renase

Buscar

PRODUCTOS RENASE

	Propoleo Extracto US \$2.0
	Propoleo en Miel US \$3.0
	Sangre de Drago US \$5.0

Detalles de los Productos

Si se desea ver las características de los productos se debe presionar sobre el producto deseado y automáticamente se desplegará la información del producto.



The screenshot shows a mobile application interface for 'Renase'. At the top, the status bar displays '3G', signal strength, battery, and the time '6:17 AM'. The app title 'Renase' is in a grey header. Below it, a small image of a yellow bottle is next to the product name 'Propoleo Extracto' in bold. The price '\$2.0 c/u' is shown to the right. The 'Presentacion:' section lists 'Gotero 30 ml.'. The 'Propiedades:' section describes it as a 'Antibiótico natural, anti-inflamatorio y anestésico.'. A link 'Conocer detalles sobre el Producto:' points to 'http://www.renase.com'. Below this is a form with the label 'Ingrese la cantidad que desea comprar:' and an empty input field. A 'Comprar Producto' button is to the right of the input field. At the bottom, a note states: 'El stock del producto es de 12 unidades, cualquier pedido mayor a esta cantidad será entregado en cuatro días.'

Búsqueda de los Productos

Si se desea buscar un producto se debe presionar en el Edit Text que se encuentra en la parte superior e ingresar el producto a buscar y automáticamente se desplegará el producto que coincida.



Iniciar Sesión

Se debe presionar el menú en la pantalla donde se listan los productos, aparecerá el botón “Iniciar Sesión” como se muestra a continuación.



Al presionar "Iniciar Sesión" se desplegará la pantalla de Login donde se deberá ingresar el nombre de usuario y contraseña.



The image shows a mobile application interface for a login screen. At the top, there is a status bar with icons for signal strength, battery, and the time 6:23 AM. Below the status bar is a header with the text "Renase". The main content area contains two input fields: "Usuario:" followed by a text input field with a cursor, and "Contraseña:" followed by a password input field. Below the password field is a button labeled "Ingresar". At the bottom of the screen, there is a link that says "Regístrate en el Sistema".

Registro de Cliente

Para registrarse en el sistema se debe presionar sobre “Registrarse en el Sistema” y a continuación se desplegará la pantalla de registro donde se debe ingresar la información requerida.



The image shows a mobile application interface for a registration form. At the top, the status bar displays 'EG', signal strength, and the time '6:40 AM'. Below the status bar, the title 'Renase' is visible. The main heading is 'Ingrese la siguiente información'. The form consists of several text input fields, each with a red label to its left: 'Nombre:', 'Cedula o RUC:', 'Ciudad:', 'Dirección:', 'Correo:', 'Telefono:', 'Celular:', 'Nombre de Usuario:', and 'Clave:'. The 'Nombre:' field is currently selected, indicated by an orange border. At the bottom of the form, there are two buttons: 'Registrarse' and 'Cancelar'.

Una vez ingresada la información se debe presionar sobre “Registrarse” para terminar con el registro.

Actualizar Datos y Cambio de Contraseña

Se debe presionar el botón menú en la pantalla de lista de productos y seleccionar “Mi Cuenta”.



En la pantalla “Mi Cuenta” se debe seleccionar “Actualizar Datos” o “Cambios de Clave”



A continuación se desplegará la pantalla según la opción elegida anteriormente en la cual se ingresará los cambios deseados.



Renase

Actualización de datos

Nombre: Erik Mafla

Ciudad: Quito

Dirección: Cap. Rafael ramos 531 y Lt

Córeo: erickdm20@hotmail.com

Telefono: 2418942

Celular: 095828005

Actualizar Ahora Cancelar



Renase

Ingrese su nueva clave:

Cambiar Ahora Cancelar

Realizar Pedidos

Una vez seleccionado el producto, se debe ingresar la cantidad no mayor a 100 unidades y posteriormente se debe presionar “Comprar Producto”.



The screenshot shows a mobile application interface for purchasing a product. At the top, the app name 'Renase' is visible on the left, and the user's name 'Usuario: emafia' is on the right. The product name 'Propoleo en Miel' is displayed in a large, bold font, accompanied by a small image of the product bottle. The price is listed as '\$3.0 c/u'. Below the product name, the presentation is specified as 'Presentacion: Frasco 125 ml.' and the properties are listed as 'Propiedades: Problemas respiratorios y afecciones a la garganta.' A link to 'Conocer detalles sobre el Producto:' is provided as '<http://www.renase.com>'. A section for entering the quantity to purchase is labeled 'Ingrese la cantidad que desea comprar:' and contains a text input field with the number '23' and a 'Comprar Producto' button. At the bottom, a note states: 'El stock del producto es de 34 unidades, cualquier pedido mayor a esta cantidad será entregado en cuatro días.'

A continuación se desplegará una alerta que nos indicara si se quiere continuar comprando o quiere finalizar la compra, si se selecciona “Continuar Comprando” se volverá a la pantalla de lista de producto y se selecciona “Finalizar Pedido” se desplegará automáticamente la pantalla de detalle del pedido.




Al presionar sobre el producto se puede editar la cantidad o se puede eliminar el producto de la lista.



The screenshot shows a mobile application interface for a product named "Propoleo en Miel". At the top, the status bar displays "7:27 AM" and various icons. The app header is "Renase" with the user name "Usuario: emafia" on the right. The product name "Propoleo en Miel" is displayed in bold, next to a small image of the product and its price "\$3.0 c/u". Below this, the presentation is listed as "Presentacion: Frasco 125 ml." and the properties as "Propiedades: Problemas respiratorios y afecciones a la garganta." A link to "http://www.renase.com" is provided under "Conocer detalles sobre el Producto:". A section titled "Ingrese la nueva cantidad:" contains a text input field with the number "34" and a "Guardar los Cambios" button. A note below states: "El stock del producto es de 34 unidades, cualquier pedido mayor a esta cantidad será entregado en cuatro días." At the bottom, there is a button labeled "Eliminar el Producto de mi Pedido".

Renase

Usuario: emafia

 **Propoleo en Miel** \$3.0 c/u

Presentacion:
Frasco 125 ml.

Propiedades:
Problemas respiratorios y afecciones a la garganta.

Conocer detalles sobre el Producto:
<http://www.renase.com>

Ingrese la nueva cantidad:

34

El stock del producto es de 34 unidades, cualquier pedido mayor a esta cantidad será entregado en cuatro días.

Revisar los Pedido Pendientes

En la pantalla Mi Cuenta se debe seleccionar la opción “Pedidos Realizados” y se desplegara la pantalla de todos los pedidos realizados según la fecha y que estén pendientes.

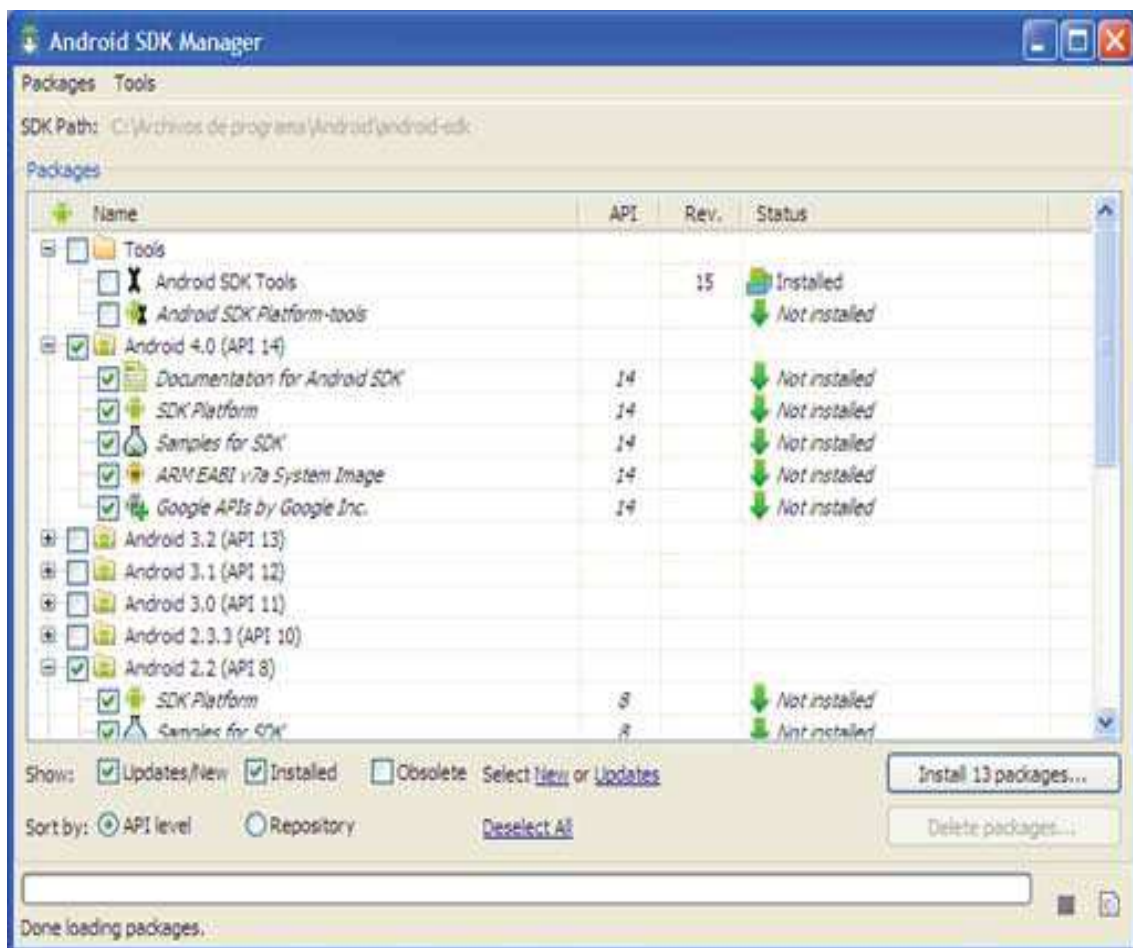


La presionar sobre cualquier pedido se desplegará automáticamente su detalle.

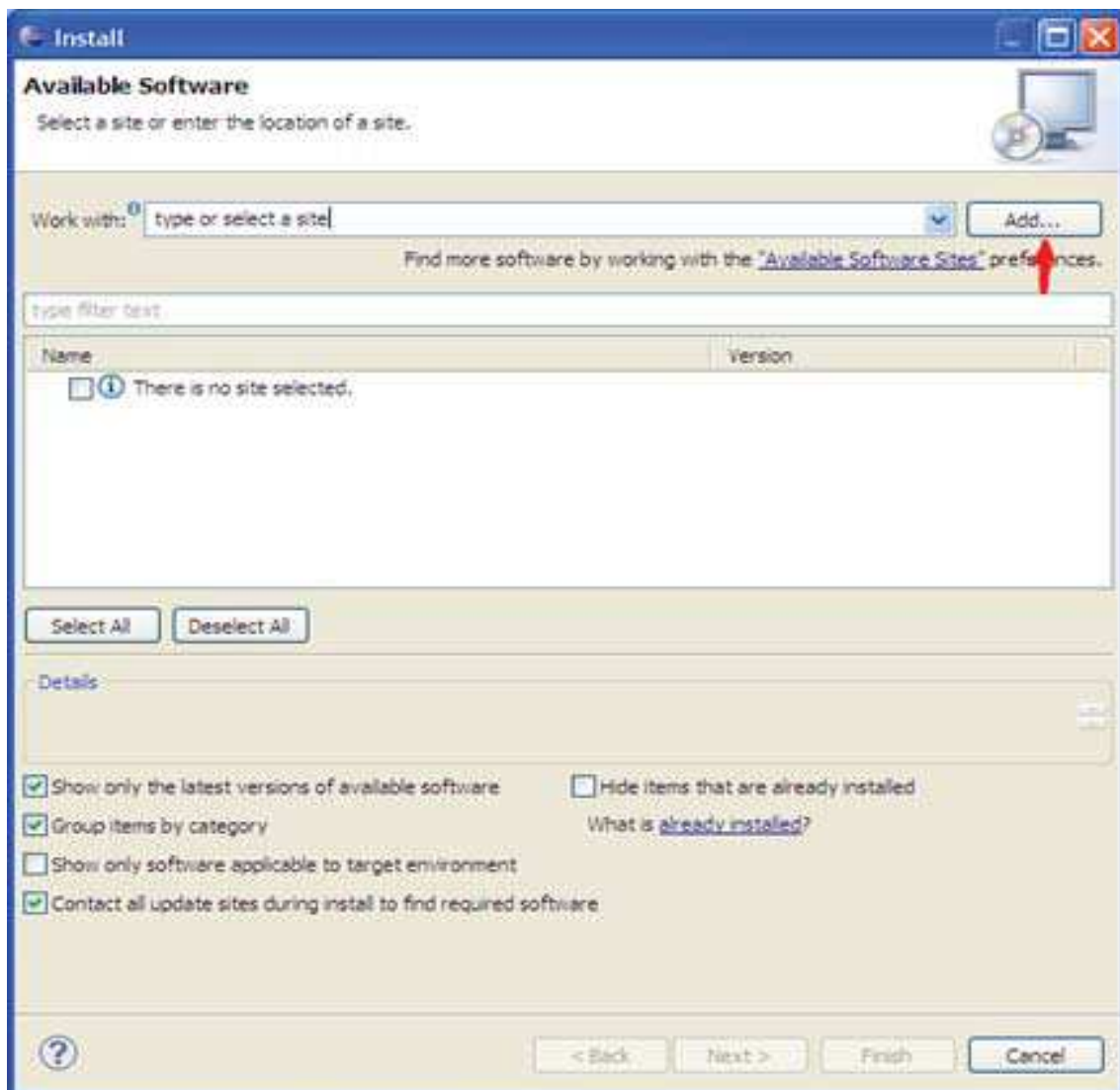


Instalar y preparar entorno para comenzar a desarrollar en Android

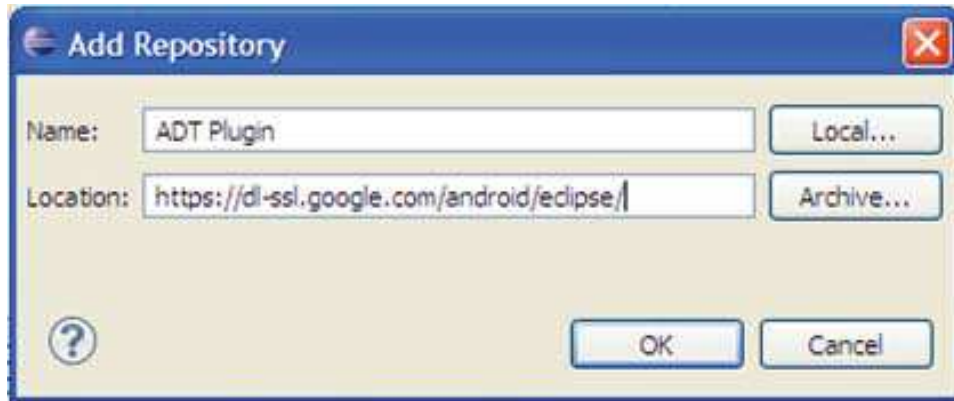
1. Descargar en SDK de Android. (Descargar el que corresponda al sistema operativo).
2. Instalar el SDK previamente descargado (ejecutar como administrador en Windows 7).
3. Ejecutar el SDK Manager tras la instalación, e instalar los “platforms” correspondientes a las versiones de Android en las que piensas desarrollar.



1. Instalar el Plug-In ADT para Eclipse (en caso de que vayas a trabajar con Eclipse). A continuación los pasos para instalarlo.
2. Primero que todo se debe tener instalado Eclipse. Se recomienda utilizar la versión Eclipse de Motorola Motodev Studio.
3. Iniciar Eclipse
4. Help → Install New Software
5. Clic en “Add” en la esquina superior derecha



6. En la ventana “Add Repository” escriba “ADT Plugin” como nombre y la siguiente URL como locación: <https://dl-ssl.google.com/android/eclipse/>



7. Haz Clic en “Ok”

Nota: Si tienes problemas intenta colocando en la URL de la locación “http” en vez de “https”.

8. El repositorio se cargará y aparecerá un elemento con una casilla de verificación al lado. Clic en la casilla de verificación para activarla.



9. Clic en “Next”.

10. En la siguiente ventana veras una lista de las herramientas que serán instaladas. Clic en “Next”.

11. “Lea” y acepte los Acuerdos de Licencia o License Agreements. Clic en “Finish”

Nota: Si te aparece una advertencia de seguridad, Clic en “Ok”



12. Cuando la instalación se complete. Reinicie Eclipse.
13. Cuando se inicie Eclipse, se iniciará un asistente pidiendo que instalemos el SDK de Android con sus respectivos Platforms. Como éste paso ya lo ejecutamos previamente, seleccionamos la segunda opción y escribimos la ruta donde instalamos nuestro Android SDK. Se instala por defecto en “C:\Archivos de programa\Android\android-sdk. (Instartius Corporation, 2012)

Este procedimiento fue utilizado para la instalación del entorno de desarrollo Android, sobre el cual se desarrolló la aplicación para la empresa Renase, utilizando el API 2.1 del SDK de Android y la versión modificada de eclipse Motodev Studio.

MYSQL

MySQL es un popular y gratuito sistema de gestión de bases de datos relacionales creado por MySQL LAB y en la actualidad desarrollado por Oracle Corporation como software libre.

Las características más importantes de este gestor de base de datos son:

- MySQL puede ser instalado en diferentes plataformas como Ubuntu o Windows.
- MySQL tiene los siguientes APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso de Multi-threaded mediante Kernel lo cual hace que pueda usar múltiples procesadores.
- Usa tablas de búsqueda b-tree con compresión de índices las cuales son muy rápidas.
- Implementa las funciones SQL.
- Soporta grandes Bases de Datos.
- Utiliza sockets para TCP/IP para conectarse desde cualquier plataforma.
- Utiliza un sistema de seguridad de privilegios y contraseñas y permite la verificación basada en host.

- Es Open Source.
- Es posible replicar las bases de datos.
- Cuenta con un sistema propio de administración el cual cuenta con los módulos SQL Development, Data Modeling y Server Administrator. (Oracle Corporation, 2011)

Como se puede observar MySQL es una herramienta de gestión de base de datos muy poderosa que puede ser usado por cualquier sistema para administrar sus bases de datos relacionales, con múltiples ventajas ya que además de ser un software gratuito contiene características similares a bases de datos pagadas.