

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN - MANAGUA

RECINTO UNIVERSITARIO RUBÉN DARÍO

DEPARTAMENTO DE COMPUTACIÓN



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

Tema:

Programación de Aplicaciones Móviles

Tema delimitado:

Aplicación Móvil bajo la plataforma Android para informe de Calificaciones y resultados de admisión de los Estudiantes de la Universidad Nacional Agraria, primer semestre año 2014.

Integrantes:

Br. Carlos Lorenzo Quezada Morales

Br. Gloria del Carmen López Hernández

Br. Marlon de Jesús Álvarez Alemán

Tutor: Lic. Roberto Ezequiel Morales Suárez



Dedicatoria

Dedico este proyecto mi hijo Axel Alejandro, a mi familia y a mis padres, por su paciencia y apoyo incondicional; a nuestros amigos y compañeros, por motivarnos con su energía; y demás personas que directa o indirectamente hicieron posible la culminación de esta etapa.

Gloria López Hernández



Dedicatoria

Dedico el presente proyecto a mi madre que siempre deseó lo mejor para mí futuro, además a mi familia que han sido la roca en que baso mis fuerzas para sostenerme en pie y seguir adelante; a las personas que incondicionalmente me han apoyado en el transcurso de mi camino, las cuales han permitido la culminación de mis estudios académicos.

Carlos Quezada Morales



Dedicatoria

Primeramente a Dios por haberme permitido llegar hasta este punto y colmarme de salud y sabiduría, ser el manantial de vida y obsequiarme lo necesario para seguir adelante día a día para lograr mis objetivos y metas propuestas.

A mi madre por apoyarme en todo momento, por aconsejarme siempre, inculcarme valores y motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor de madre.

A mi hermano, por apoyarme económicamente en mis estudios profesionales, por enseñarme que con esfuerzo y dedicación puedo conseguirlo todo.

A mis maestros por su gran apoyo y motivación para la culminación de nuestros estudios profesionales; por transmitirme los conocimientos obtenidos y llevado pasó a paso en el aprendizaje.

A mis amigos por apoyarme y estar conmigo de manera incondicional.

Marlon Álvarez Alemán



Agradecimientos

Con la realización de este proyecto se pone punto y final a todo un largo camino que ha sido la carrera de Licenciatura en Computación, mediante estas líneas queremos agradecer todo el apoyo que hemos tenido para conseguir finalizar.

Primeramente queremos dar las gracias a Dios por nuestra salud y por darnos fortaleza para salir adelante cada día.

A nuestros padres, por todo su amor y apoyo incondicional.

A Lic. Roberto Ezequiel Morales Suárez, tutor del proyecto, por todos sus consejos y tiempo invertido en ayudarnos a terminar con éxito el fin de nuestra carrera.

Y a las demás personas que hicieron posible llevar a buen término este trabajo de grado.



Índice

Contenido

I. Introducción.....	1
II. Objetivos	2
General	2
Específicos	2
III. Justificación	3
IV. Desarrollo (marco teórico).....	4
1. Sistema operativo	4
2. Sistemas Operativos móviles.....	4
2.1. iPhone IOS.....	5
2.2. Windows Mobile	6
2.3. BlackBerry	¡Error! Marcador no definido.
2.4. Symbian.....	9
2.5. Android.....	10
2.5.1. Definición de Android	10
2.5.2. Historia de android.....	11
2.5.3. Evoluciones en las Versiones de Android.....	12
2.5.3.1. Android 1.0: Apple pie.....	12
2.5.3.2. Android 1.5: Cupcake.....	13
2.5.3.3. Android 1.6: Donut.....	13
2.5.3.4. Android 2.0: Eclair	15
2.5.3.5. Android 2.1	16
2.5.3.6. Android 2.2 Froyo	17
2.5.3.7. Android 2.3 Gingerbread	18
2.5.3.8. Android 3.0: Honeycomb.....	19
2.5.3.9. Android 4.0: Ice CreamSandwich.....	21
2.5.3.10. Android 4.1: JellyBean.....	23
2.5.3.11. Android 4.3	24
2.5.3.12. Android 5.0: Key Lime Pie.....	25
3. Aplicación (APP).....	26
3.1. Aplicaciones web móviles	28
3.2. Aplicaciones Nativas.....	28
3.3. Aplicaciones Híbridas	28
4. Análisis y diseño de sistemas.....	29



5. Análisis y diseño estructurado	29
6. Análisis y diseño Orientada a Objetos.	30
7. Modelo del proceso del Software	31
7.1. El modelo lineal (o modelo cascada)	31
7.2. El modelo incremental	32
7.3. El modelo de construcción de prototipo.....	32
7.4. El modelo en espiral	33
8. Base de Datos	35
8.1. Bases De Datos Relacionales.....	36
8.2. Diseño de Base de Datos	36
8.3. Modelo Relacional.....	37
8.4. Normalización	37
8.4.1. Primera forma Normal (1FN).....	38
8.4.2. Segunda Forma Normal (2FN)	39
8.4.3. Tercera Forma Normal (3FN).....	40
8.4.4. Tercera Forma Normal Boyce-Codd (3FNBC)	40
8.4.5. Dependencias Multi-valuadas y la Cuarta Forma Normal (4FN)	40
8.5. MySQL	41
9. Entorno de Desarrollo	43
9.1. Eclipse	43
9.2. Programación Orientada a Objetos (POO).....	45
9.3. Servicio Web.....	47
10. UML (Lenguaje Modelado Unificado).....	50
10.1. Diagramas	51
10.2. Diccionarios de Datos.....	58
11. Universidad Nacional Agraria (UNA).....	58
11.1. Requisitos de Admisión.....	60
11.2. Requisitos para realizar pre-matrícula:.....	62
11.3. Sistema de Evaluación.....	64
V. Diseño Metodológico.....	71
1. Tipo de investigación a realizar	71
2. Universo y muestra	71
2.1. Universo.....	71
2.2. Muestra	72
3. Obtención de los datos	72
3.1. La observación	72
3.2. La entrevista.....	72
3.3. La encuesta.....	73
4. Metodología utilizada para la producción de software	73



5. Fases del modelo incremental	74
5.1. Fase en el desarrollo del sistema	74
5.1.1. Incremento 1	74
5.1.2. Incremento 2	76
VI. Análisis de Resultado	78
1. Encuesta	78
2. Herramientas y Tecnologías	82
3. Descripción de la Aplicación	82
3.1. Consulta de Resultados de Notas Semestrales	83
3.2. Consulta de resultados de Admisión	83
4. Estudio de Factibilidad	83
4.1. Factibilidad Técnica	83
4.2. Factibilidad Económica	85
4.3. Factibilidad operacional	86
5. Diagrama UML	87
5.1. Casos de usos	87
5.2. Descripción de Casos de Usos	88
5.3. Diagrama de Secuencia	90
5.4. Diagrama de Clase	92
5.5. Diagrama Entidad-Relación	94
6. Diccionario de datos	96
7. Pantallas de la Aplicación	101
7.1. Pantallas con Datos	105
8. Pruebas de la Aplicación.	107
VII. Conclusiones	108
VIII. Recomendaciones	109
IX. Bibliografía	110
X. Anexos	112



Resumen

En el presente trabajo se abordó el análisis y diseño de una Aplicación móvil, que permitirá a los estudiantes de primer ingreso de la Universidad Nacional Agraria consultar sus resultados de admisión. A su vez accederá a las calificaciones de asignaturas semestrales de los estudiantes activos.

Seguidamente se muestra los objetivos planteados y la documentación necesaria para la realización de la aplicación.

La recopilación de la información se desarrolló, realizando visitas, encuestas personales y telefónicas a los estudiantes y personal de UNA. Esto permitió la obtención de datos que identifican las necesidades y expectativas de los usuarios finales de la aplicación móvil.

Posteriormente se muestra cada una de las fases de la evolución del proyecto, así como los requerimientos necesarios que hicieron posible la realización de dicha aplicación móvil; se realizó un estudio de factibilidad con los requerimientos mínimos de la adquisición de los equipos para su desempeño.

Para finalizar se detalla las conclusiones y recomendaciones de la aplicación en su etapa final.



I. Introducción

Es un hecho que la revolución móvil ha cambiado por completo los hábitos y tendencias de usuarios y consumidores. Los nuevos dispositivos móviles se han convertido en una ventana de acceso permanente al universo de la red, y desde la cual se accede desde cualquier lugar y en cualquier momento.

Evidentemente, este escenario ha provocado que medios, como empresas y marcas, adopten nuevas estrategias y tecnologías para llegar a este tipo de audiencias móviles.

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier lugar, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño.

El propósito de este trabajo es aprovechar la tecnología móvil para desarrollar una aplicación móvil Android que permita realizar consultas de calificaciones a los estudiantes de la Universidad Nacional Agraria en el primer semestre del año 2014.



II. Objetivos

General

Desarrollar una aplicación móvil bajo la plataforma Androide que permita a los estudiantes hacer consulta de notas y resultados de admisión en la Universidad Nacional Agraria.

Específicos

- ❖ Determinar las condiciones tecnológicas presentes en la Universidad Nacional Agraria.
- ❖ Analizar los procesos principales relacionados al trámite de nota y consulta de resultados de admisión de la Universidad
- ❖ Diseñar las interfaces correspondientes para la consulta de Calificaciones y resultado de admisión en la Universidad Nacional Agraria.
- ❖ Programar la aplicación bajo la plataforma Android utilizando el lenguaje de programación java y los aspectos básicos del desarrollo orientada a objetos.



III. Justificación

Las redes móviles existentes en Nicaragua están extendiendo la cobertura a nivel nacional, lo cual es positivo para los estudiantes que poseen un teléfono con estas aplicaciones; tomando en cuenta la ventaja, nos hemos propuesto desarrollar esta aplicación móvil Android para que los estudiantes puedan realizar sus consultas académicas desde la comodidad de su hogar.

Actualmente, la Universidad Nacional Agraria posee una estructura de Red muy sofisticada, en la que se puede acceder a ella en las diferentes zonas wifi que la Universidad ha establecido para los estudiantes y visitantes. La wifi posee un mismo nombre, esta metodología le permite al estudiante a permanecer en línea sin estar buscando otro punto de conexión. Con esta ventaja la aplicación se podrá tener acceso desde cada una de las zonas wifi.

Esta permitirá a los estudiantes acceder a información de gran relevancia de forma rápida y sencilla, con el requisito único de contar con internet en el móvil, consultar ésta información desde el sitio web en algunos casos resulta complicado por las muchas opciones que se muestran, además que desde un móvil no es posible acceder por su gran cantidad de scripts y otros recursos que lo vuelven pesado.

Con esta aplicación lo que se busca es optimizar el tiempo y crear comodidad a los estudiantes para que realicen sus consultas y agilizar sus gestiones académicas en el menor tiempo posible.



IV. Desarrollo (marco teórico)

1. Sistema operativo

Un sistema operativo es un programa que administra el hardware de una computadora. También proporciona la base para los programas de aplicación y actúa como un intermediario entre el usuario y el hardware de la computadora. Los sistemas operativos para main frame están diseñados principalmente para optimizar el uso del hardware.

Los sistemas operativos de las computadoras personales (pc), soportan desde complejos juegos hasta aplicaciones de negocio. Los de las computadoras de mano están diseñados para proporcionar un entorno en el que el usuario pueda interactuar fácilmente con la computadora para ejecutar programas. Por tanto, algunos sistemas operativos se diseñan para ser prácticos, otros para ser eficientes y otros para ser ambas cosas.

2. Sistemas Operativos móviles.

Un sistema operativo móvil o SO móvil, es un sistema que controla un dispositivo móvil al igual que los PCs utilizan Windows o Linux entre otros. Sin embargo, los SO móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Muchas personas se encuentran ante un dilema al momento de elegir un smartphone cuando se trata de buscar información de los sistemas operativos de los equipos.



2.1. iPhone IOS

Desarrollado por la compañía Apple, está incluido en la mayoría de sus dispositivos (iPhone, iPods y Tabletas). El iPhone se basa en OS X, se lanzó en el año 2007, que a su vez es una variante de Unix, uno de los sistemas operativos más poderosos en el mundo de la informática.

Es el sistema operativo de los iPhone. Ha demostrado ser un excelente producto pero limitado, única y exclusivamente a los terminales producidos por Apple lo que limita su crecimiento en el mercado y obliga al consumidor a depender de un solo fabricante.



El nombre de la interfaz es CocoaTouch. Se trata de un API para la creación de programas para el iPad, iPhone y iPod Touch de la compañía Apple Inc. CocoaTouch proporciona una capa de abstracción al sistema operativo iOS.

CocoaTouch se basa en el set de herramientas que proporciona el API de Cocoa para crear programas sobre la plataforma Mac OS X.



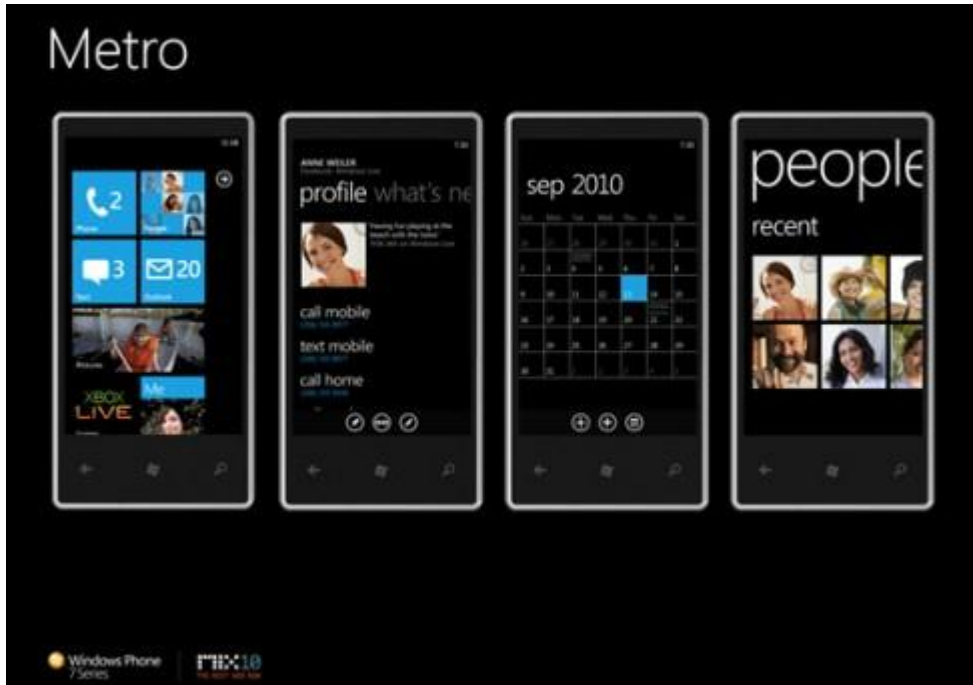
Herramientas para desarrollar aplicaciones basadas en CocoaTouch se incluyen en el SDK de iOS.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles. Los elementos de control consisten en deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado o horizontal).



2.2 Windows Mobile

Para las personas que buscan una oficina móvil además de todas las herramientas que brinda Microsoft Windows (Office, MSN Messenger, etc.), Windows Mobile es la opción correcta. El sistema operativo desarrollado por Microsoft, es la adecuación de la conocida plataforma Windows, utilizada en la mayoría de las PCs alrededor del mundo. En su versión móvil se encuentran la mayoría de las aplicaciones ya conocidas como la “barra de inicio” y la “barra de tareas”, además de programas de productividad como Outlook, Microsoft Office, y de entretenimiento como MSN Messenger, o Windows Media Player.



Metro es el nombre del sistema de diseño de las interfaces de Windows Phone. El nombre es debido a que utiliza el mismo lenguaje visual que aparece en aeropuertos y en la señalización de las redes de metro. La meta de utilizar Metro es facilitar que los usuarios lleguen al contenido que deseen gracias al uso de una tipografía conocida por medio de un diseño simple y claro logrando una experiencia de usuario fluida.

Interfaz "toque y arrastre", fácilmente podemos movernos en cualquier orden que deseemos. Al deslizar la derecha nos lleva a una lista de todas sus aplicaciones. Si deseas mover una de sus aplicaciones a la pantalla de inicio rápido, simplemente presiónala y deja así por un momento y en el menú seleccionar "añadir a inicio rápido". Cuando estás en una aplicación, verás una fila de otras opciones de menú en la parte superior de la pantalla. Si desea volver al menú anterior, sólo tiene que pulsar la tecla de la flecha en la parte superior de la interfaz.

Son muchos los que defienden a Android a capa y espada, bien por su libertad o por su filosofía. Es por eso que se ha convertido en uno de los grandes en el



sector de los sistemas operativos móviles. Pero esto no echa para atrás a otras empresas con sus propios SO que deciden competir, como por ejemplo es el caso de Windows con su Windows Phone 8.



2.3 BlackBerry

Blackberry tuvo un duro 2011 debido a que estaban orientados más al mundo de los negocios sin embargo, estas personas comenzaron a irse con otros fabricantes causando una baja significativa en las ventas de las compañías.

Blackberry ha estado modificando su visión y los modelos cuentan con todo lo que se puede esperar de un móvil moderno, no sólo para el trabajo, sino también para uso personal.

Desarrollado por la empresa canadiense RIM (Research In Motion) usa un kernel propio, que al igual que Android, tiene un motor Java, y aunque han mejorado la interfaz notablemente, suele mostrar algunas limitaciones propias de su edad.





El sistema operativo más nuevo de esta compañía se llama BlackBerry 7 e incluye varias características y servicios nuevos muy interesantes, así como algunas mejoras en las características existentes.

2.4 Symbian

Symbian es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y Handhelds de PSION.

El 24 de junio de 2008, Nokia decidió comprar Symbian, adquiriendo el 52% restante de las acciones de la compañía, tras un acuerdo con el resto de socios. El objetivo era establecer la Fundación Symbian y convertir este sistema operativo en una plataforma abierta.

Symbian OS está hecho para las personas que buscan un sistema operativo intuitivo y estable, además de reproductor multimedia, cámara fotográfica, video y música. El sistema operativo también trabaja en una plataforma de código abierto y lo desarrolla Symbian Foundation desde hace poco menos de 10 años. Ahora se puede disfrutar de stream de música, bajar aplicaciones, revisar correo electrónico y mucho más.



Symbian Belle es el nombre de la nueva versión e interfaz Symbian, una de las tantas mejoras que trae Symbian Belle, es el poder utilizar las aplicaciones para la función NFC que los equipos ya traían pero que no funcionaban. Además, cuenta con nuevos widgets y la posibilidad de personalizar mucho más el equipo.

2.5. Android

2.5.1. Definición de Android

Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. El sistema operativo de Google, Android, tiene apenas 6 años y en este tiempo lo hemos visto evolucionar de una manera realmente



impresionante, mediante las versiones que periódicamente saca al mercado Android es un claro ejemplo de todos los cambios que el sistema ha sufrido.

2.5.2. Historia de android

Allá por octubre del año 2003, Andy Rubin, Rich Miner, Nick Sears y Chris White daban forma a Android Inc. En sus inicios, únicamente trascendió que la actividad de la empresa se centraba en “el desarrollo de software para teléfonos móviles”.

Android Inc. pasó casi dos años trabajando “en la sombra”, hasta que Google comenzó a “reclutar” a fuerza de talonario a algunas “startup” (término que se refiere a nuevas compañías con un futuro prometedor) del sector móvil, con la clara intención de replicar su éxito de la Web en el futuro de las telecomunicaciones inalámbricas.

La fecha clave para llegar a entender mejor el éxito de Android es el 5 de noviembre de 2007. Ese día se fundaba la OHA (Open Handset Alliance), una alianza comercial de 35 componentes iniciales liderada por Google, que contaba con fabricantes de terminales móviles, operadores de telecomunicaciones, fabricantes de chips y desarrolladores de software. El mismo día se dio a conocer por vez primera lo que hoy conocemos como Android, una plataforma de código abierto para móviles que se presentaba con la garantía de estar basada en el sistema operativo Linux.

Ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos software de código abierto.

Aunque no fue hasta un año después, en Octubre de 2008 cuando lo vimos por primera vez funcionando en un HTC Dream. Veía la luz en los USA un móvil con la primera versión final de Android, la 1.0. El modelo G1 de HTC quedará para la historia como el iniciador de este gigante llamado Android.



Una de las claves de la popularidad de Android es que, como Linux, es una plataforma de código abierto, lo que permite a fabricantes, operadores y desarrolladores dar mayor funcionalidad a sus smartphones. Cinco millones de terminales al mes salen con Android, lo que no deja de preocupar a los que tienen su propio sistema cerrado, como Nokia (Symbian), Apple (iOS) o RIM (Blackberry). Además, Android es un sistema gratuito y multiplataforma; por multiplataforma entendemos que el sistema operativo puede ser usado en distintas plataformas, y por plataforma entendemos que es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos. Android es gratuito al ir instalado en cualquier dispositivo móvil.

2.5.3. Evoluciones en las Versiones de Android

2.5.3.1 Android 1.0: Apple pie



Lanzado el **22 de octubre de 2008**, el HTC Dream también conocido por entonces como Google Phone, fue el primer dispositivo en incorporar el sistema operativo de Google.



Este incluyó la primera versión de la AndroidMarket, un Navegador Web, soporte para mensajes de texto SMS y MMS, discador para llamadas, y una aplicación para tomar fotos que no contaba con los ajustes de blancos y resolución.

Por otro lado, se incluyó una aplicación capaz de acceder a los servidores de correo de terceros, con soporte para los estándares POP3, IMAP4, y SMTP.¹⁴ que era capaz de sincronizarse con aplicación de Gmail, Google y Google Calendar. Tampoco faltó el

reproductor de archivos multimedia que por entonces no era capaz de reproducir video

Por último, cabe destacar que Android 1.0 ofreció desde sus inicios el soporte para WiFi , Bluetooth y el popular sistema de notificaciones que aparecen en la barra de estado, con la posibilidad de configurar alertas por ringtone, LED o vibración.



2.5.3.2. Android 1.5: Cupcake



Con la introducción de android 1.5 el 30 de abril de 2009, empezamos a oír el nombre de cupcake en referencia a la primera actualización importante del sistema operativo de Google.

Esta actualización le dio un poco más pulido a Android en algunas áreas, pero sus principales características fueron la introducción del teclado virtual en la pantalla (todavía malo) y la

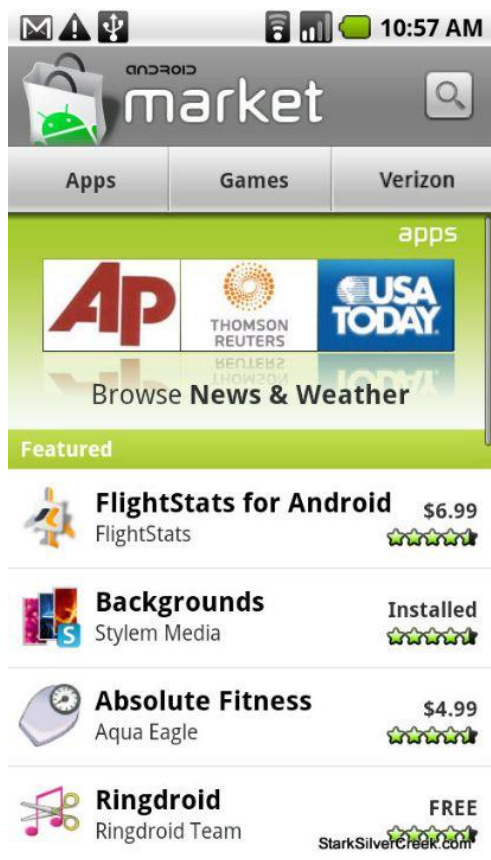
posibilidad de insertar widgets.

Además, se incluyeron otras funciones bastante demandadas por los usuarios como copiar y pegar en el navegador, la grabación de vídeo y reproducción en formatos MPEG-4 y Android 1.63GP, la capacidad de subir videos a YouTube directamente, transiciones animadas entre las pantallas, la opción de auto-rotación, auto-sincronización y soporte para Bluetooth A2DP y AVRCP.

2.5.3.3. Android 1.6: Donut



Luego vino el conocido Donut. Esta versión fue en realidad una pequeña actualización, pero dotada con un cuadro de búsqueda mejorado, cámara y aplicación de galería y una renovada **AndroidMarket**.



Con la llegada de Donut apareció el soporte para redes CDMA haciendo que Android llegara a Estados Unidos y Asia. Pero quizá la mejora más significativa, fue la posibilidad de correr el sistema operativo en múltiples resoluciones de pantalla y relaciones de aspecto, a raíz de esta actualización es cuando podemos disfrutar hoy en día de pantallas con resolución QVGA, HVGA, WVGA, FWVGA, QHD y 720p.

La barra de búsqueda, que inicialmente tenía sólo para buscar en la web, ahora le permitirá al usuario indagar en otros lugares dentro del dispositivo; el cambio más notable en **Donut** fue el AndroidMarket que en ese momento renovó su diseño con colores verde y blanco, frente a la mirada gris y negro de las versiones anteriores.

La aplicación de la cámara también, vio una remodelación y si bien no era la más bonita, era todavía un paso adelante respecto a lo que estábamos trabajando con anterioridad.

Ahora un usuario puede seleccionar fácilmente la grabación de vídeo sin salir de la aplicación, así como los ajustes que estaban ocultos en la parte izquierda de la pantalla en una barra de menú deslizante lateral. Según el sitio de desarrolladores de Android, la nueva aplicación de la Cámara era un 39% más rápida y el tiempo entre disparo y disparo fue mejorado en un 28%.

Lo único lamentable de la actualización de **Android 1.6**, fue que no muchos dispositivos la recibieron y la mayoría como el DroideEris o Hero saltaron directamente de **Android 1.5** y **Android 2.1**.



2.5.3.4. Android 2.0: Eclair



Lanzada el 26 de octubre del 2009, la actualización de **Android 2.0 Eclair** debuto en noviembre de ese mismo año en los Motorola Droid y se trató de un hito muy importante para la plataforma que dio paso al crecimiento exponencial y la atención de las masas.

AndroidEclair, nos sorprendió con su integración social, permitiendo sincronizar los contactos de Facebook y más tarde Twitter, que les permitió a sus usuarios tener todos sus contactos en las redes sociales en un solo lugar.

En cuanto a la interfaz de usuarios, también se realizaron mejoras que recayeron básicamente en las animaciones, en las transiciones y su fluidez general.

Sin embargo, esta actualización no se detuvo allí y nos trajo un puñado de funciones nuevas para la cámara, como el zoom digital, modo de escena, balance de blancos, efectos de color y el enfoque macro. No obstante tendríamos que esperar hasta **Froyo** para que la aplicación de la cámara se puliera lo suficiente, como para darnos la experiencia agradable que tenemos hoy.

Por otro lado, el teclado virtual de **Android** fue mejorado también con el soporte multitouch, y el diccionario de sugerencias ampliado, que incluía los nombres de nuestros contactos.

El navegador de Android también recibió una actualización, sorprendiendo con la nueva función doble toque para el zoom, lo que permitía ampliar la foto sin necesidad de que los usuarios tengan que depender exclusivamente de los botones más y menos en la parte inferior de la pantalla y el navegador comenzó a



dar soporte para HTML5. Finalmente, Google Maps recibió el servicio de navegación GPS gratuito.

2.5.3.5. Android 2.1

Representa la segunda etapa en la evolución de Eclair con su introducción en el NexusOne. Ésta trajo consigo los fondos de pantalla animados e interactivos, siendo ahora hasta 5 escritorios de serie, en lugar de los 3 que mostraban las versiones anteriores.

El NexusOne, fue también el primer teléfono que extendiera las capacidades de voz existente, encontrados en versiones anteriores de Android, dando al usuario la opción de traducir la voz en texto en cualquier campo.

También introdujo algunos efectos 3D en el sistema operativo, entre los que podemos encontrar el ícono para lanzar las aplicaciones, ahora basta con un golpecito en el lanzador de aplicación, para revelar su contenido mientras que antes era necesario arrastrar hacia arriba la pestaña.

La galería de fotos también vio una importante remodelación en 3D, con la ayuda de Cooliris que logró una de las más bonitas aplicaciones integradas para el sistema operativo hasta la fecha.

2.5.3.6. Android 2.2 Froyo



Lanzada el **20 de mayo de 2010**, **Android 2.2 Froyo**, fue una de las actualizaciones que consagró al sistema operativo como la competencia de **iOS**



4 de Apple, dotando a los terminales Android con un notable incremento de velocidad en todo el sistema, tanto en sus aplicaciones como en la navegación de Internet.

Froyo incorpora el motor de Java V8 y ofrece a los usuarios un aumento de velocidad gracias al compilador JIT que permite iniciar las solicitudes más rápido y mejorar el rendimiento general del sistema.

A su vez, **Android 2.2** incluye la posibilidad de hacer **tethering** es decir, compartir la conexión 3G a través del wifi del teléfono con otros dispositivos, con la posibilidad de convertir tu móvil en un hotspot.

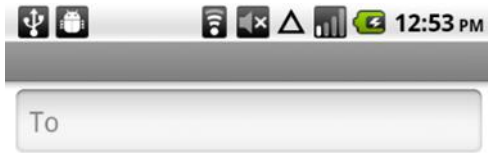
Una característica que los usuarios habían estado esperando durante años, se hace realidad en **Android 2.2** y se trata del soporte para Adobe Flash, tanto para el navegador de Internet como para reproducir contenidos multimedia a través del Flash Player.

Una vez que un dispositivo se ha actualizado para **Froyo**, el reproductor de Flash se puede encontrar en el AndroidMarket y tiene muy buenos resultados, demostrando así, que la tecnología multimedia puede ejecutarse en un teléfono móvil.

Por último cabe destacar otras características incluidas como la opción para mover las aplicaciones a las tarjetas microSD, una pantalla de inicio ligeramente modificada, nuevos widgets, mejoras en la galería de fotos, un puñado de



características de Exchange, así como la APIcloud-to-device que le permite enviar páginas web y direcciones de Google Maps, desde tu ordenador al teléfono.

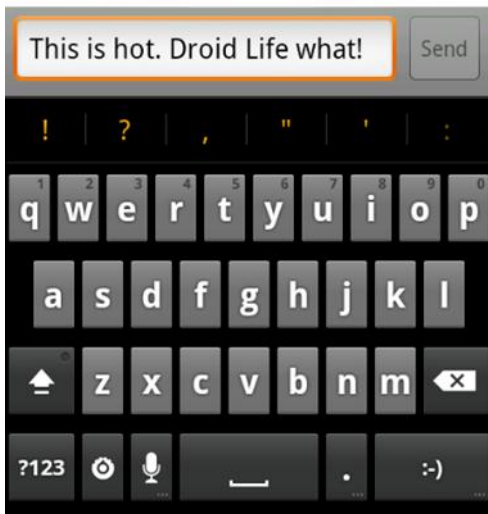


2.5.3.7. Android 2.3 Gingerbread



El **6 de diciembre de 2010** Google presentó de forma oficial **Android 2.3 Gingerbread**, una actualización que se materializaría con el lanzamiento del Nexus S.

Gingerbread incorporó una gran cantidad de novedades tanto estético con una renovada interfaz de usuario, con incrementos de velocidad y simpleza, y se preparó para la llegada de los smartphones de doble núcleo, al cambiar al sistema de archivos EXT4 y de pantallas más grandes con el soporte para resoluciones WXGA y mayores.



Del lado del usuario, una de las características más notables fue el nuevo teclado virtual que simplificó la entrada de

texto y permitió una edición más rápida, gracias a la nueva disposición de las teclas y la función para corregir palabras ya ingresadas, con sugerencias del diccionario o la opción de cambiarlas mediante voz.

Sin dudas, la adquisición de BlindType tuvo que ver en este sentido y le permitió a Google implementar con características, como permitir el deslizamiento al teclear, asistencia en la escritura, ajustes personalizados al estilo de escritura del comprador y el “multitouchkey-chording”, que facilite al usuario ingresar rápidamente números y símbolos presionando Shift+ y ?123+, sin necesidad de cambiar los métodos de entrada manualmente.

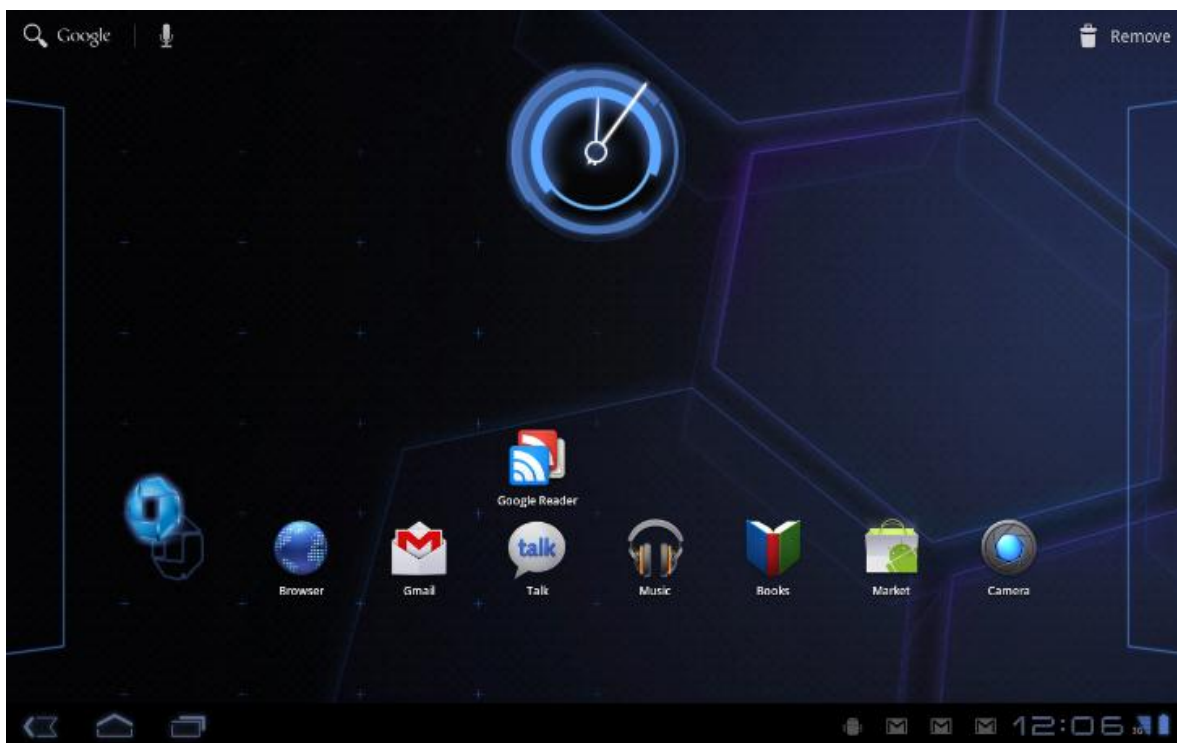
Por otro lado, Android 2.3 incorporó toda una gama de funciones que permiten manejar el dispositivo con la voz en lo que se denominó Voice Actions, además esta actualización incorporó el soporte para llamadas VoIP/SIP; el protocolo



basado en una interfaz inalámbrica con el que se podrán pagar diferentes cuentas desde el móvil llamado NFC y una gestión de la energía mejorada.

A su vez, Gingerbread incluyó una nueva pestaña de “Running” dentro de Manage Apps, que muestra la lista de aplicaciones activas junto con la capacidad y memoria que están consumiendo cada una de ellas.

2.5.3.8. Android 3.0: Honeycomb



El **22 de febrero de 2011**, Google comenzó a desdoblarse el sistema operativo con la actualización de **Android 3.0 Honeycomb** y su correspondiente SDK, algo que tendría poca vida debido al alto costo que supone mantener dos plataformas separadas.

Basado en el kernel 2.6.36.50 de linux, Honeycomb llegó por primera vez en las tablets Motorola Xoom 24 y sus principales características fueron una renovada interfaz de usuario con una nueva barra de sistema en la parte inferior de la pantalla, que permitía el acceso rápido a notificaciones, estados y botones de



navegación suavizados y el Action Bar que permitía el acceso a opciones contextuales, navegación, widgets y otros tipos de contenido desde la parte superior.

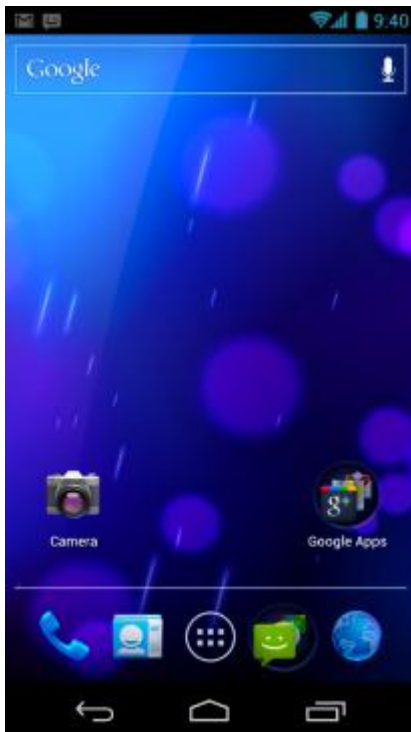
Además, se agregó una nueva interfaz de contactos dividida en dos paneles, algo que también caló en la interfaz de correo, para simplificar la visualización y organización de correos, permitiendo a su vez seleccionar uno o más mensajes.

Por otro lado la actualización de Honeycomb, trajo un teclado re-diseñado para pantallas de gran tamaño y se simplificó la función multitarea con una opción que permitió acceder a las aplicaciones recientes que se mostraban en una lista, con imágenes para reconocerlas fácilmente.

El navegador también tuvo cambios con la llegada de las pestañas que reemplazaron a las ventanas, la característica de auto completado al ingresar texto y un nuevo modo incógnito que permitió la navegación de forma anónima como el navegador web.

Por último, cabe mencionar el soporte para microprocesadores multi-núcleo, la aceleración de hardware, la posibilidad de encriptar todos los datos del usuario y mejoras en el uso de HTTPS, gracias a la incorporación de SNI.

2.5.3.9. Android 4.0: Ice CreamSandwich

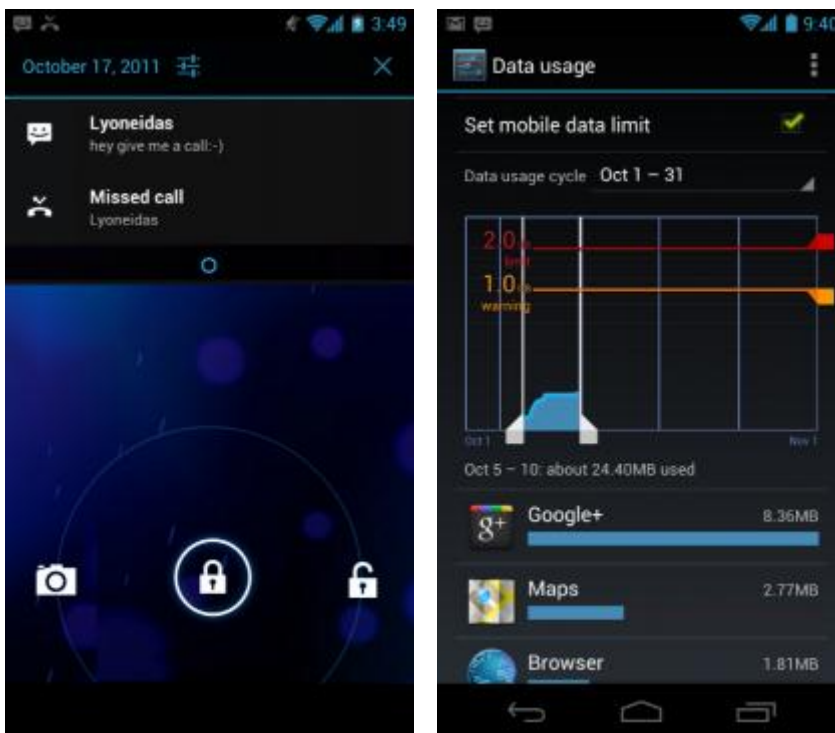


La llegada de **Android 4.0 Ice CreamSandwich** el **19 de octubre de 2011**, significó un importante paso en la evolución de Android que no sólo vio renovada casi por completo su interfaz de usuario con el nuevo diseño Holo, sino que volvió a integrar el sistema operativo en sus versiones para Tablets y Smartphones. La nueva interfaz de usuario se mostró como la evolución y perfeccionamiento de las ideas de Android 3.0, dándole un poco de esa mirada limpia y futurista. Además Google construyó su propia fuente denominada Roboto y en lugar de botones de hardware, el sistema operativo ofreció sus propios botones virtuales de atrás, inicio, y los botones de aplicaciones recientes en la pantalla también para los smartphones. Además, el dock de aplicaciones ahora incluyó una nueva sección para mostrar los widgets de forma separada donde son listados de forma similar a las aplicaciones y se simplificó la posibilidad de crear carpetas, con estilo de arrastrar y soltar.



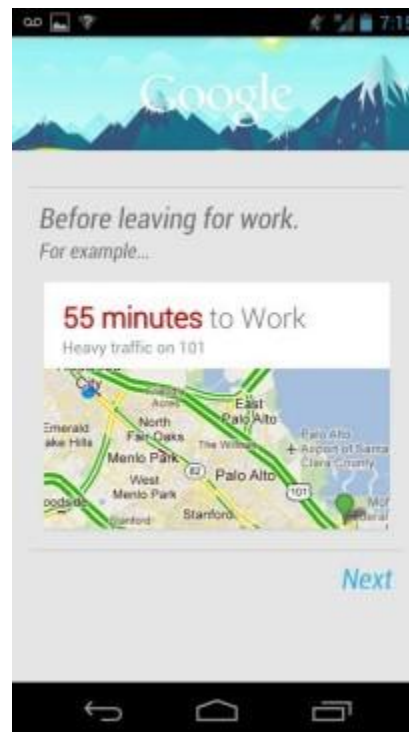
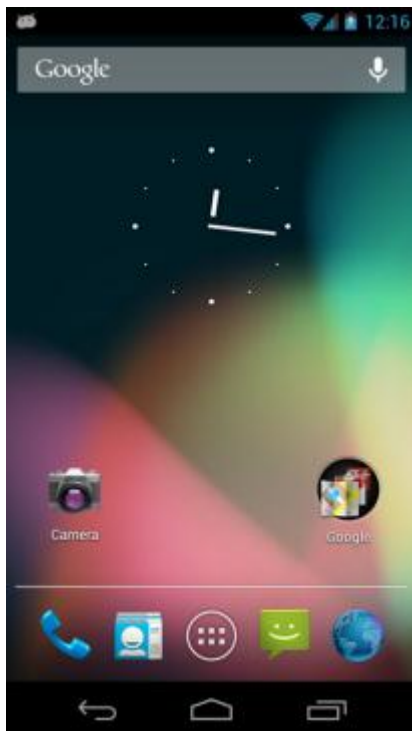
Pero no todo en Android 4.0 tuvo que ver con el diseño, Google incluyó algunas mejoras que hoy usamos a diario, como la posibilidad de acceder a las aplicaciones directamente desde la pantalla de bloqueo y Google Chrome como navegador por defecto que permitió abrir hasta a 15 pestañas y realizar la sincronización automática con los marcadores de la versión de escritorio.

Otra de las grandes novedades fue el desbloqueo facial, característica que permite desbloquear los Smartphone usando el software de reconocimiento, algo que luego sería muy criticado por su dudosa seguridad; y una nueva sección que nos permitió controlar de forma nativa el consumo de datos de todo el equipo y configurar límites para evitar exceder nuestro plan, así como cerrar aplicaciones que están usando datos en segundo plano.



Otras aplicaciones originarias mejoradas fueron la cámara que ya no mostró retardo en el obturador y permitió realizar ajustes sobre el time-lapse, además se mejoró la aplicación People para integrarla con las redes sociales y permitir la actualización de estados e imágenes en alta resolución, se incorporó de forma natural la posibilidad de tomar screenshots presionado los botones de bloqueo y de bajar volumen y se mejoró la funcionalidad copiar-pegar.

2.5.3.10. Android 4.1: JellyBean



Y así es como llegamos a los tiempos modernos donde JellyBean aún resuena como la última actualización importante del sistema operativo de Google que dicho sea de paso, fue presentada el **27 de junio de 2012** y llegó al mercado el 13 de julio con el Nexus 7, el primer tablet de Google.

El objetivo primordial de **Android JellyBean** fue mejorar la estabilidad, funcionalidad y rendimiento de la interfaz de usuario, para lo cual se implementó el núcleo de linux 3.0.31 y una serie de mejoras en lo que se llamó ProjectButter que permitió aumentar hasta 60 FPS las transiciones en la interfaz de usuario, dando una experiencia realmente fluida.

Sin embargo, esto no fue todo lo que trabajó JellyBean, ya que Google mejoró notablemente la barra de notificaciones, una de las características que distinguió a Android desde sus inicios.



Google Now, fue otra de las grandes características de esta actualización, que junto al KnowledgeGraph y la búsqueda por voz mejorada, permitió superar ampliamente a Siri, el asistente de Apple, ya que fue capaz de reconocer y predecir nuestros intereses en función del historial de búsquedas.

Los widgets fueron desde los inicios de Android una de sus características distintivas y en esta actualización recibieron cierta atención, ya que se ajustan automáticamente al tamaño de la pantalla, cuando son grandes para caber en ella

Finalmente, otra de las mejoras estuvieron centradas en la entrada de texto, donde por un lado, fue mejorada la entrada por voz que ya no requirió tener una conexión a internet para utilizarla, dado que el intérprete se encuentra dentro del dispositivo y el teclado predictivo que reconoce hasta cuatro idiomas, es capaz de adivinar la próxima palabra que vamos escribir.

2.5.3.11. Android 4.3





El pasado 24 de Julio Google presentó Android 4.3 JellyBean, una pequeña actualización que introdujo algunas mejoras de seguridad y rendimiento en el sistema operativo para darle mayor fluidez.

De este modo se han introducido mejoras en la representación de formas redondeadas y texto, y la velocidad en que se muestran las imágenes así como el soporte para OpenGL ES 3.0, Bluetooth Smart (o Bluetooth LE) y optimizaciones en vsync timing y el triple buffering.

La aceleración de hardware 2D ahora optimiza el flujo de comandos de dibujo, convirtiéndolo en un formato GPU más eficiente, reorganizando y uniendo operaciones de dibujo, lo que se suma al procesamiento multiproceso que le permite al procesador utilizar hilos múltiples a través de los diferentes núcleos del CPU en determinadas tareas.

Además, Android 4.3 ha incorporado el soporte para perfiles restringidos que permite crear ambientes separados para cada usuario en el mismo dispositivo, haciendo que el administrador sea capaz de determinar qué acciones puede realizar cada usuario, cómo descargar aplicaciones de Google Play, realizar compras in-app, jugar a determinados juegos, acceder a ciertas apps, etc.

Pero eso no es todo, Android 4.3 JellyBean también añade un nuevo marco de DRM modular, soporte para codificación VP8 integrado, mejoras en el soporte RTL, mejoras en seguridad gracias a SELinux, Google Play Games, mejoras en la entrada de texto, nueva interfaz de la cámara, autocompletado al marcar un número de teléfono, mejor gestión de la batería y nuevas versiones de las GApps como Gmail, Hangouts, etc.

2.5.3.12. Android 5.0: Key Lime Pie

Hasta ahora es todo lo que sabemos de Android y la verdad es que no me aventuraría a sacar muchas conclusiones. Tras la falta de Android 4.3 en el Google I/O, el supuesto Android 5.0 Key Lime Pie, aparece aún más lejos en el

horizonte y resulta difícil predecir cualquier cosa tras el paso de Android a manos de Sudar Pichai que según sus declaraciones, tiene intenciones de profundizar en la experiencia de usuario.



3. Aplicación (APP)

Una aplicación (también llamada app), es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que aunque todas las aplicaciones son programas, no todos son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina



así a aquél que ha sido creado con un fin determinado, para realizar tareas concretas. No se consideraría una aplicación por ejemplo, un sistema operativo, ni una suite, pues su propósito es general.

Las aplicaciones nacen de alguna necesidad concreta de los usuarios y se usan para facilitar o permitir la ejecución de ciertas tareas en las que un analista o un programador ha detectado una cierta necesidad. Pero las aplicaciones también pueden responder a necesidades lúdicas, además de laborales (todos los juegos, por ejemplo, son considerados aplicaciones). Suele decirse que para cada problema hay una solución y en informática, para cada problema hay una aplicación.

Naturalmente, el campo de las aplicaciones es tan extenso y las funciones tan dispares, que se han creado numerosas clasificaciones, según varios criterios. Así por ejemplo, dependiendo del uso que se le vaya a dar, podemos tener aplicaciones profesionales o personales. También podemos hablar de aplicaciones para Windows, Linux, Android, Apple, PC, móvil, de acuerdo el sistema operativo o la plataforma que vayamos a usar. Según su finalidad, pueden ser de entretenimiento, negocios, empresariales, de sonido, gráficas, de edición, web, didácticas... e incluso según la rama de las ciencias en el que se usen podemos decir que hay aplicaciones de ingeniería, telecomunicaciones, matemáticas...

Las aplicaciones pueden ser desde pequeñísimos programas de apenas unas líneas de código, hasta grandes obras de ingeniería informática, con miles de horas de trabajo detrás. El tamaño sin embargo, no define el éxito de una aplicación, sino justamente que **cubra las necesidades del usuario**

Ahora bien, para el desarrollo de app para teléfonos inteligentes, hay diferentes opciones de aplicaciones que se deben ser comprendidas para poder elaborar la aplicación:



- ❖ Aplicaciones web móviles
- ❖ Aplicaciones móviles nativas
- ❖ Aplicaciones híbridas

3.1. Aplicaciones web móviles

El desarrollo móvil web es el desarrollo de aplicaciones web regulares, pero optimizadas para ser visualizadas desde un dispositivo móvil o Tablet. Por definición, estas aplicaciones serán accedidas utilizando el navegador que viene por defecto dentro de los dispositivos. Que esto viene siendo como consultar una página web, que dependiendo del tamaño se adecue la estructura de la página, como es el caso de la tecnología responsive web design (Diseño web sensible o adaptativo).

3.2. Aplicaciones Nativas

Por otra parte, el desarrollo móvil nativo es el adelanto de aplicaciones que serán instaladas en el sistema de archivo de cada dispositivo y serán distribuidos por los mercados de aplicaciones móviles, como el AppStore (IOS) o el Play Store (Android).

3.3. Aplicaciones Híbridas

Luego de hablar de las dos opciones de aplicaciones, por qué no hacer una mezcla de ambas opciones, a esto es lo que se le conoce como aplicaciones móviles híbridas, que han tenido mayor relevancia gracias a PhoneGap.

Este tipo de aplicaciones permite el uso de tecnologías multiforma como HTML, JavaScript y CSS pero permite acceder a gran parte de los dispositivos y sensores del teléfono. Buena parte de la infraestructura es tipo web y la



comunicación con los elementos del teléfono, se hace mediante comunicaciones tales como PhoneGap. Un buen ejemplo de aplicaciones híbridas es Facebook.

La idea de utilizar aplicaciones híbridas, es poder llegar no sólo a una plataforma, sino que se pueda extender a varias, sin necesidad de estar sacando nuevas versiones de la aplicación cada cierto tiempo.

4. Análisis y diseño de sistemas

El análisis y diseño de sistemas, se refiere al proceso de examinar la situación de una empresa con el propósito de manejarla con métodos y procedimientos más adecuados.

En pocas palabras análisis especifica qué es lo que el sistema debe hacer. El diseño establece cómo alcanzar el objetivo. Ciertamente, todo sistema de información debe presentar salidas en base a entradas de datos y procesos, lo que nos dice que si deseamos entender todo lo que le ocurre a los datos antes de llegar al usuario como información .Es decir, antes de ser interpretado por el usuario final, debemos utilizar metodologías que permiten ver los sistemas en base a sus procesos, por lo menos en sistemas de procesado por lotes o secuencial; Un ejemplo de ello es la metodología estructurada. Existen muchas metodologías pero esta es la más arraigada debido a su antigüedad. Recordemos que hace apenas dos décadas, los computadores no soportaban el multitasking (procesamiento multitarea), lo que limitaba a procesar una pantalla a la vez, esto sólo permitía sistemas secuenciales donde cada tarea en procesamiento comenzaba cuando la anterior ya había terminado por completo.

5. Análisis y diseño estructurado

Permite al analista conocer un sistema o proceso (actividad) en una forma lógica y manejable, al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle pertinente. El objetivo que persigue el análisis estructurado, es organizar las tareas asociadas con la determinación de requerimientos para



obtener la comprensión completa y exacta de una situación dada.

Está compuesto por:

- ❖ Símbolos gráficos: sirven para identificar y describir los componentes de un sistema y las relaciones entre estos.
- ❖ Diccionarios de datos: Descripciones de todos los datos utilizados en el sistema, pueden ser manual o automatizado.
- ❖ Descripciones de procesos y procedimientos: emplean técnicas y lenguajes que permiten describir actividades del sistema.
- ❖ Reglas: Estándares para describir y documentar el sistema en forma correcta y completa.
- ❖ Análisis de flujo de datos: Estudia el empleo de los datos en cada actividad, documenta los hallazgos con diagramas de flujo de datos.

6. Análisis y diseño Orientada a Objetos.

La programación orientada a objetos (OOP) promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo. En el desarrollo de software: la falta de portabilidad del código y reusabilidad, que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas, tiene tres características básicas: debe estar basado en objetos, clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos, muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un



dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo. Es una cosa real o abstracta que está formada por un conjunto de otros objetos y que poseen una estructura lógica para una determinada función.

7. Modelo del proceso del Software

Para resolver problemas reales de una industria, debe incorporar una estrategia de desarrollo que acompañe al proceso, métodos y capas de herramientas; esta estrategia se llama modelo de proceso o paradigma de ingeniería de software.

7.1. El modelo lineal (o modelo cascada)

Es el más antiguo de todos los modelos de Ingeniería del Software. El modelo lineal presenta una estructura secuencial (de ahí el nombre de Modelo en cascada) formada por seis fases o etapas:

- ❖ Análisis del Sistema
- ❖ Análisis de Requisitos de Software
- ❖ Diseño
- ❖ Codificación
- ❖ Prueba
- ❖ Mantenimiento

Las fases incluyen dentro de sí determinadas tareas que clasifican de una forma clara el trabajo a realizar.

El desarrollo de las fases, como he mencionado antes, se produce de manera secuencial. Una vez se produce el análisis tanto del Sistema como de los requisitos del software demandado por el cliente, (fases en las que la intervención del cliente es absolutamente necesaria), se procede a la fase de diseño de la



arquitectura global del software. Un diseño elaborado de forma cuidadosa llevará a una rápida codificación. Tras haber traducido el programa a un lenguaje comprensible para el ordenador, se comprueban los elementos de forma individual y más tarde de manera homogénea (todos los sistemas a la vez). Una vez entregado el software al cliente, la fase de Mantenimiento comprenderá las actualizaciones y las correcciones de errores que sean necesarias en el programa.

El Modelo en cascada no permite retroceder (más tarde analizaremos las ventajas e inconvenientes de todos los modelos en común), por lo que se hace estrictamente necesario que al final de cada fase el analista de sistemas o, en su caso, el programador, verifique y valide todo el trabajo realizado, ya que un error no detectado a tiempo podría perjudicar gravemente la fecha de entrega del software a nuestro cliente.

7.2. El modelo incremental

Es una evolución del modelo de cascada; viene a suplir el problema de no poder retroceder en las fases de desarrollo del software. Es, por tanto, un modelo no secuencial.

El funcionamiento es sencillo, comienza con el análisis de los requisitos, tras el cual se prepara un primer diseño. La novedad de este modelo respecto del anterior, es la introducción de iteraciones para “bifurcar” diseños. Es decir, este modelo ofrece la posibilidad de comenzar un diseño, arquitectura, estructura, etc del software, que de no convencer al cliente (o al propio programador) es rechazado y se comienza con una segunda iteración (o un segundo diseño), sin necesidad de realizar un nuevo análisis de requisitos. Pueden realizarse tantas iteraciones (también llamadas incrementos) como sean necesarias.

7.3. El modelo de construcción de prototipo

Este modelo no secuencial, basado en la construcción de simulaciones o modelos ejecutables de aplicaciones más extensos, persigue un objetivo principal: la



participación directa del cliente en la construcción del software requerido. Las fases son similares a las del modelo en cascada: es necesario un análisis previo de los requisitos tanto del sistema como del cliente, se concibe la arquitectura del sistema y se realiza el diseño del software. Sin embargo, se incluye un elemento hasta ahora no utilizado, que consiste en el diseño rápido de un prototipo que se mostrará al cliente para que evalúe el trabajo realizado.

El prototipo es una versión reducida del programa completo, es una “fachada virtual” que mostramos al cliente (que carece de la posibilidad de ser utilizada de la forma en que lo haríamos con el software final. Tras recoger los requisitos tanto del cliente como del sistema, se comienza con el diseño rápido del prototipo; el diseño completo obedece al previo diseño de pequeños prototipos específicos para funciones individuales. Más tarde, estos diseños serán unidos en uno sólo.

Después, se procede a la construcción del mismo. Éste prototipo es el que mostraremos al cliente para que lo evalúe y considere cambios en él, aunque no se trate de una versión definitiva.

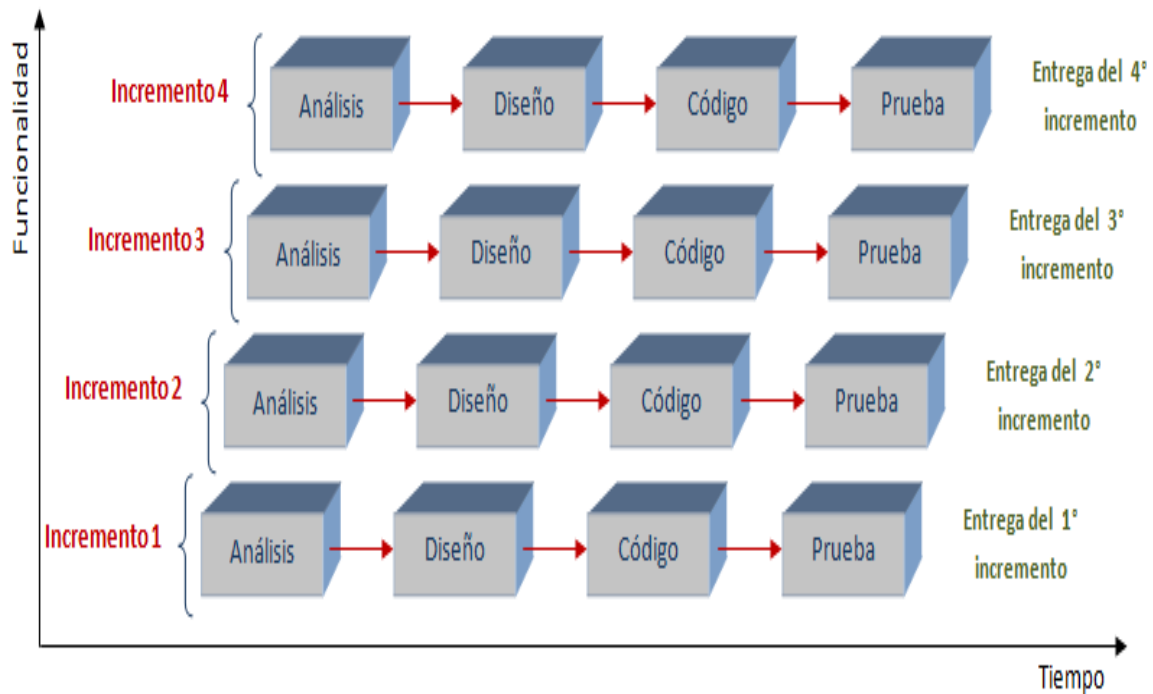
7.4. El modelo en espiral

Este modelo también no secuencial, es algo más complejo que los anteriores, aunque incluye un elemento muy útil e importante en el desarrollo del software: análisis de riesgos. El modelo en espiral concreta cuatro fases:

- ❖ Planificación
- ❖ Análisis de Riesgos
- ❖ Ingeniería (Construcción del prototipo)
- ❖ Evaluación por el cliente

Si ésta última fase es afirmativa, el modelo continúa con la estructura del Ciclo de vida Clásico. Si el cliente no está satisfecho con el resultado, se cubre otra banda de la espiral y se vuelve a la primera fase (de planificación).

El modelo a utilizar en nuestro proyecto es el incremental debido a sus características de desarrollo ágil que permiten un trabajo más corto y preciso. A continuación información detallada del modelo a desarrollar:



Figura# 1 Modelo Incremental fuente: google

El modelo incremental combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos, aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software. Por ejemplo, el software de tratamiento de textos desarrollado con el paradigma incremental, podría extraer funciones de gestión de archivos básicos y de producción de documentos en el primer incremento; funciones de edición más sofisticadas y de producción de documentos en el segundo incremento; corrección



ortográfica y gramatical en el tercero; y una función avanzada de esquema de página en el cuarto. Se debería tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos.

Cuando se utiliza un modelo incremental, el primer incremento a menudo es un producto esencial. Es decir, se afrontan requisitos básicos, pero muchas funciones suplementarias (algunas conocidas, otras no) quedan sin extraer. El cliente utiliza el producto central (o sufre la revisión detallada). Como un resultado de utilización y/o de evaluación, se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones, y características adicionales. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.

El modelo de proceso incremental, como la construcción de prototipos y otros enfoques evolutivos, es iterativo por naturaleza. Pero a diferencia de la construcción de prototipos, el modelo incremental se centra en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones «incompletas» del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

El desarrollo incremental es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en la fecha límite que se haya establecido para el proyecto. Los primeros incrementos se pueden implementar con menos personas.

8. Base de Datos

¿Qué es una Base de Datos?

Es un conjunto de datos acumulados en un dispositivo de almacenamiento masivo el cual se encuentra disponible en forma simultánea a un número de usuarios automatizados y en un tiempo pertinente, por su parte el software facilita la



comunicación de los usuarios con la base de datos por medio de un lenguaje de consultas y en donde se garantiza la integridad y seguridad de los datos.

8.1. Bases De Datos Relacionales

Existen muchos tipos de base de datos en función del modo en que almacenan y acceden a la información que guardan: relacional, jerárquica, en la red, orientada a objetos, etc. Ejemplos de gestores de bases de datos relacionales o RDBMS (Relational Database Management System) hay muchos: MySQL, SQLite, Oracle, Informix, SyBase, Microsoft SQL Server, Postgres, Msq, etc.

Básicamente, un gestor de base de datos relacionales almacena los datos en Tablas, cada una de las cuales está formada por filas(o registros), y estas a su vez, están formadas por columnas (o campos), antes de definir una tabla, hay que normalizarla, proceso que consiste en evitar redundancia, es decir, que la información esté duplicada ya que, si hubiese que cambiar un dato repetido, habría que cambiar varias veces.

8.2. Diseño de Base de Datos

Una única base de datos relacional puede contener cualquier número de tablas, esto plantea las preguntas de, cuántas tablas debe tener una base de datos y de qué campos debe tener cada una. Dichas decisiones son la base del diseño de datos.

Un buen diseño hace que resulte fácil trabajar con la base de datos y ofrece suficiente flexibilidad como para admitir los requisitos futuros. Un diseño pobre obstaculiza las operaciones select, proyect y join, haciendo que la base de datos resulte poco flexible y sea difícil trabajar con ella.



8.3. Modelo Relacional

Las ideas fundamentales del modelo de datos relacional aparecieron en forma pública, en un artículo de Edward Codd de 1970, desde ese momento, se ha establecido como el principal modelo para las aplicaciones de procesamiento de datos, debido a su simplicidad que facilita el trabajo del programador en comparación con otros modelos como el de red y jerárquico.

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se les asigna un nombre exclusivo, donde cada fila de la tabla representa una relación entre un conjunto de valores, dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y de relación, por tal razón es que el eje principal del modelo relacional reposa sobre el concepto de relación. (Silberschatz, 2006)

8.4. Normalización

El proceso de normalización es un método propio del modelo relacional y consiste en descomponer las relaciones originales en otras más pequeñas con el fin de eliminar una serie de anomalías de almacenamiento y manipulación que se pueden dar en las relaciones iniciales y que conformarían la futura base de datos relacional, así las relaciones además de ser más simples y estables, son más fáciles de mantener que otras estructuras de datos.

Entre los beneficios que se pueden dar en una base de datos correctamente normalizada, se encuentran los siguientes:

- ❖ Reducir los problemas asociados con la supresión e inserción de tuplas.
- ❖ Reducir el tiempo asociado con modificaciones de las tuplas.
- ❖ Identificar problemas potenciales que pueden requerir un análisis adicional
- ❖ Mejorar la información para la toma de decisión, referente a la organización física de los datos.



El proceso de normalización se compone de una serie de seis etapas llamadas formas normales, a continuación se definirán cada una de ellas:

8.4.1. Primera forma Normal (1FN)

Se refiere a una representación de una relación en la cual los atributos son diferentes y los valores de cada uno de ellos son componentes atómicos, en esta 1FN lo que se busca es evitar los grupos repetitivos

Dentro de esta se encuentran las dependencias multi-valuadas que pueden establecer asociaciones entre varios atributos de una misma relación. Así, los valores de algunos atributos en una relación pueden determinar en forma única, el conocimiento de los valores de otros de la misma relación. Estas dependencias traducen reglas semánticas que, como se sabe, son muy comunes en el montaje del mundo real (Gonzalez, 1996)

Axiomas de Inferencia de Armstrong

Sean X, Y, Z subconjuntos de atributos de una relación R en donde se verifican las dependencias funcionales $X \rightarrow Y$ y $Y \rightarrow Z$. Entonces, las siguientes reglas se cumplen:

Reflexividad	$X \rightarrow X$ se verifica siempre
Aumento	$X \rightarrow Y \Rightarrow X \cup Z \rightarrow Y$
Transitividad	$\{X \rightarrow Y, Y \rightarrow Z\} \Rightarrow X \rightarrow Z$

A partir de este conjunto de reglas se pueden inferir otras como las siguientes:

Sean X, Y, Z y W subconjuntos de atributos de una relación R. Entonces se verifican las siguientes relaciones:



Unión	$\{X \rightarrow Y \text{ y } X \rightarrow Z\} \Rightarrow X \rightarrow Y \cup Z$
Descomposición	$X \rightarrow Y \Rightarrow X \rightarrow Z \text{ con } Z \subseteq Y$
Pseudo-transitividad	$\{X \rightarrow Y \text{ y } Y \cup Z \rightarrow W\} \Rightarrow X \cup Z \rightarrow W$

Cerradura

Es el conjunto de dependencias funcionales que se pueden derivar de un conjunto de dependencias funcionales F. se denota por F^+ , es decir

$$F^+ = \{f / F \vdash f\}$$

Saturación

Sea F un conjunto de dependencias funcionales. Se le llama saturación de un conjunto X con respecto a F, y se denota por X^+ , al conjunto de atributos que son determinados por las dependencias funcionales que se derivan de F

Teorema de descomposición

Sea un esquema de relación R (X, Y, Z), con X, Y y Z conjuntos de atributos de R, tal que la dependencia funcional $X \rightarrow Y$ se verifica en R. Entonces la relación R se descompone en las relaciones $R_1 = R[X, Y]$ y $R_2 = R[X, Z]$, $R = R_1 * R_2$.

8.4.2. Segunda Forma Normal (2FN)

Para comprender lo que es la 2FN debemos conocer el concepto de dependencia parcial, González explica que sea R un esquema de relación, X una llave de R y A un atributo no llave. Se dice que el atributo A depende parcialmente de X si se verifica que $Y \rightarrow A$, en donde Y es un subconjunto propio de la llave X.

Una relación R se dice que se encuentra en segunda forma normal (2FN), si se



encuentra en 1FN y si ningún atributo no llave depende parcialmente de la llave primaria.

8.4.3. Tercera Forma Normal (3FN)

González explica que sea $R(X, Y, Z)$ un esquema de relación, en donde X, Y, Z son subconjuntos de atributos. Se dice que Z es transitivamente dependiente de X si existe Y tal que se dan las siguientes condiciones:

- ❖ Se verifica $X \rightarrow Y$
- ❖ No se verifica $Y \rightarrow X$
- ❖ Se verifica $Y \rightarrow Z$

Se dice que una relación R se encuentra en tercera forma normal (3FN) si se encuentra en 2FN y no existe una dependencia transitiva entre atributos no llave

8.4.4. Tercera Forma Normal Boyce-Codd (3FNBC)

Una relación se encuentra en tercera forma normal Boyce-Codd (3FNBC) si todos los atributos son determinados solo por llaves, es decir, si cada vez que $X \rightarrow A$, $A \notin X$, se verifica en R , entonces X contiene una llave en R .

8.4.5. Dependencias Multi-valuadas y la Cuarta Forma Normal (4FN)

Según González dice que una dependencia multi valuada existe entre dos conjuntos de atributos X y Y , y se denota por

$$X \twoheadrightarrow Y$$

Si sólo el conocimiento de X , e independientemente de otros atributos, determina un conjunto de valores relativos a Y .

Entonces se dice que una relación $R(X, Y, Z)$ se encuentra en 4FN si se



encuentra en 3FNBC y si cada vez que una dependencia multi valuada $X \rightarrow Y$ se verifica, entonces X contiene a una llave de R.

8.5. MySQL

MySQL es un **sistema de administración de bases de datos relacional** (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribución para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL server y DB2.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuarios, administrar el sistema, proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no se ha oído nunca hablar. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de datos relacionales. (Gilfillan, 2003)

Características de MySQL

En este apartado enumeraremos las prestaciones que caracterizan a este RDBMS, así como las definiciones de diseños, limitaciones o partes del estándar aun no implementadas.

Prestaciones

- ❖ Está desarrollado en C/C++
- ❖ Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.



- ❖ La API se encuentra disponibles en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- ❖ Está optimizado para equipos de múltiples procesadores.
- ❖ Es muy destacable su velocidad de respuesta.
- ❖ Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- ❖ Cuenta con un rico conjunto de tipos de datos.
- ❖ Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- ❖ Su administración se basa en usuarios y privilegios.
- ❖ Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- ❖ Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.
- ❖ Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra “ñ”.

Conectándose con el servidor

Para conectarse con el servidor debemos asegurarnos de que éste está funcionando y que admite conexiones, sean estas locales (el RDBMS se está ejecutando en la misma que intenta la conexión) o remotas.

Adicionalmente, debemos disponer de las credenciales necesarias para la conexión. Distintos tipos de credenciales nos permitirán distintos niveles de acceso. Adicionalmente, deberemos disponer de las credenciales necesarias para la conexión. Distintos tipos de credenciales nos permitirán distintos niveles de acceso. (uoc)



9. Entorno de Desarrollo

9.1. Eclipse

Fue desarrollado originalmente por IBM Canadá como el sucesor de su familia de herramientas para VisualAge. Actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de Código abierto y un conjunto de productos complementarios, capacidades y servicios. En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como Código abierto. En 2003, la fundación independiente de IBM fue creada. (ecured)

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por sí misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java (JDT)

Mientras que la mayoría de los usuarios están felices de usar Eclipse como un IDE de Java, sus ambiciones no se detienen ahí. Eclipse también incluye el Entorno de Desarrollo de Complementos (PDE), que es de interés principalmente para los desarrolladores que quieren extender Eclipse, dado que les permite construir herramientas que se integran sin dificultades con el entorno de Eclipse. Dado que todo en Eclipse es un complemento, todos los desarrolladores de herramientas tienen un campo de juego de nivel para ofrecer extensiones a Eclipse y para proporcionar un entorno de desarrollo integrado y unificado para los usuarios.

Esta paridad y consistencia no está limitada a las herramientas de desarrollo de Java. Aunque Eclipse se escribe en el lenguaje Java, su uso no se limita al lenguaje Java. Por ejemplo, los complementos se encuentran disponibles o planificados para incluir soporte para los lenguajes de programación como C/C++ y COBOL. El marco de trabajo de Eclipse puede también utilizarse como base



para otros tipos de aplicaciones que no se relacionen con el desarrollo del software, como los sistemas de gestión de contenido. (ibm)

Ventajas de Eclipse

- ❖ El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la Plataforma de Cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.
- ❖ Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos.
- ❖ La arquitectura plug-in permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros Lenguajes de programación. (ecured)

Plataforma de cliente enriquecido

La base para Eclipse es la plataforma de cliente enriquecido (del Inglés Rich Client Platform RCP). Esta plataforma está constituida por los siguientes componentes:

- ❖ Plataforma principal - inicio de Eclipse, ejecución de plugins
- ❖ OSGi - una plataforma para bundling estándar.
- ❖ El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- ❖ Face - manejo de archivos, manejo de texto, editores de texto
- ❖ El Workbench de Eclipse - vistas, editores, perspectivas, asistentes



Características de eclipse

- ❖ Dispone de un editor de texto con resaltado de sintaxis donde puedes ver el contenido del fichero en el que estás trabajando.
- ❖ Contiene una lista de tareas y otros módulos similares.
- ❖ La compilación es en tiempo real.
- ❖ Tiene pruebas unitarias con JUnit.
- ❖ Integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests y refactorización. (ecured)

9.2. Programación Orientada a Objetos (POO)

La programación orientada a objetos o POO, es un paradigma de programación que usa los objetos en sus interacciones para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Conceptos Fundamentales

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

Clase

Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Herencia

(Por ejemplo, herencia de la clase C a la clase D) es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados"(private) también se heredan, pero



como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de POO.

Objeto

Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa). Es una instancia a una clase.

Método

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento

Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

Atributos

Características que tiene la clase

Mensaje

Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus



métodos conciertos parámetros asociados al evento que lo generó.

9.3. Servicio Web

¿Qué es un servicio Web?

Es un conjunto de protocolos y estándares que son usados para exponer información y funcionalidad en un ambiente distribuido.

Los servicios web están basados en **XML**, y las aplicaciones distribuidas acceden a ellos por Internet usando protocolos de comunicación como SOAP (Simple Object Access Protocol), el cual provee una forma estándar de simple de manejo y envío de mensajes en XML (Android curso)

Servicios Web Basados en SOAP

SOAP es el protocolo más utilizado en la actualidad para implementar servicios Web. Fue creado por Microsoft, IBM y otros, aunque en la actualidad está bajo el auspicio de la W3C.

Utiliza como transporte HTTP, aunque también es posible utilizar otros métodos, como el correo electrónico. (Android curso)

Características

- ❖ **Extensibilidad** (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
- ❖ **Neutralidad** (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS).
- ❖ **Independencia** (SOAP permite cualquier modelo de programación).

Ventajas

- ❖ Debido al uso de XML permite invocar procedimientos remotos de muchos lenguajes, por lo tanto, presenta una gran interoperabilidad.



- ❖ Al utilizar una comunicación vía HTTP es fácilmente escalable, además de ser casi siempre permitido por los cortafuegos.
- ❖ Puede ser implementado utilizando cualquier lenguaje y ejecutado en cualquier plataforma.
- ❖ Es posible utilizarlo mediante usuario anónimo y mediante autenticación.
- ❖ Es posible transmitirlo mediante cualquier protocolo de transporte capaz de transmitir texto, típicamente HTTP o SMTP. (wikipedia)

Desventajas

- ❖ Debido al uso de XML para el paso de mensajes, SOAP es considerablemente más lento que otros middleware como CORBA ya que los datos binarios se codifican como texto. Para contrarrestar este punto débil en el caso de XML con código binario incrustado, se desarrolló un método optimizado de transmisión de mensajes.
- ❖ Depende del WSDL (Web Services Description Language).
- ❖ Al contrario que Java, PHP o Python ciertos lenguajes no ofrecen un apoyo adecuado para su uso ya sea a nivel de integración o de soporte IDE. (wikipedia)

Servicio Web con Ksoap de Android

En primer lugar hay que empezar diciendo que Android no incluye en su SDK ningún tipo de soporte para el acceso a servicios web de tipo SOAP. Es por esto por lo que vamos a utilizar una librería externa para hacernos más fácil esta tarea. Entre la oferta actual, la opción más popular y más utilizada es la librería ksoap-android. Esta librería es un *fork*, especialmente adaptado para Android. Este framework nos permitirá de forma relativamente fácil y cómoda utilizar servicios web que utilicen el estándar SOAP.

KSOAP es una librería tipo cliente **SOAP** elegante, ligera y eficiente. La biblioteca es de código abierto, fácil de usar, y puede salvarlo de la molestia de la



codificación de un analizador personalizado. Es una de las bibliotecas de **SOAP** más fiables disponibles en la actualidad y se actualiza con frecuencia, lo cual es una buena señal para un proyecto de código abierto.

(programacion)

Características

❖ Analizador XML

El analizador utilizado en kSOAP es KXML un eficiente Analizador Tira tipo de analizador.

❖ De / Serializador

Esta sección es responsable de la cartografía de la representación de objetos para la representación XML y viceversa. Cuando se utiliza en complexTypes kSOAP, los objetos que la capa de serialización puede operar en implementar el KVMSerializable interfaz. Desde la plataforma J2ME (la plataforma de destino primario) no tiene reflejo, esta interfaz proporciona un mecanismo que utiliza el serializador para establecer y obtener propiedades en el objeto anónimo. Para utilizar este servicio de mapas que necesita "registrar" su "Serializable" objetos para que el motor sabe cómo asignar el nombre que encuentra en el xml para el nombre de la clase que se carga en los valores. Si los objetos no son tipos complejos o los tipos complejos no están registrados, el serializador puede en volver los objetos en SoapPrimitives o SoapObjects .

❖ Transporte

La capa de transporte proporciona el mecanismo en el que se intercambian los mensajes SOAP entre el servidor y el cliente. Se sigue trabajando para abstraer esta capa de manera que diferentes tipos de transportes pueden ser "enchufados" para las pruebas, o por otros medios de comunicación. Los mecanismos básicos de transporte que actualmente se incluyen en la biblioteca son httpTransport para plataformas J2ME y HttpTransportSE para J2SE

(programacion)



10. UML (Lenguaje Modelado Unificado)

UML, por sus siglas en inglés, (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Este permite modelar, construir, y documentar los elementos que forman un sistema de software orientado a objetos.

UML es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real (Alarcon, 2000)

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema, para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

Existen 4 tipos de elementos en UML, dependiendo del uso que se haga de ellos: Elementos estructurales, elementos de comportamiento, elementos de agrupación y elementos de anotación (Alarcon, 2000)

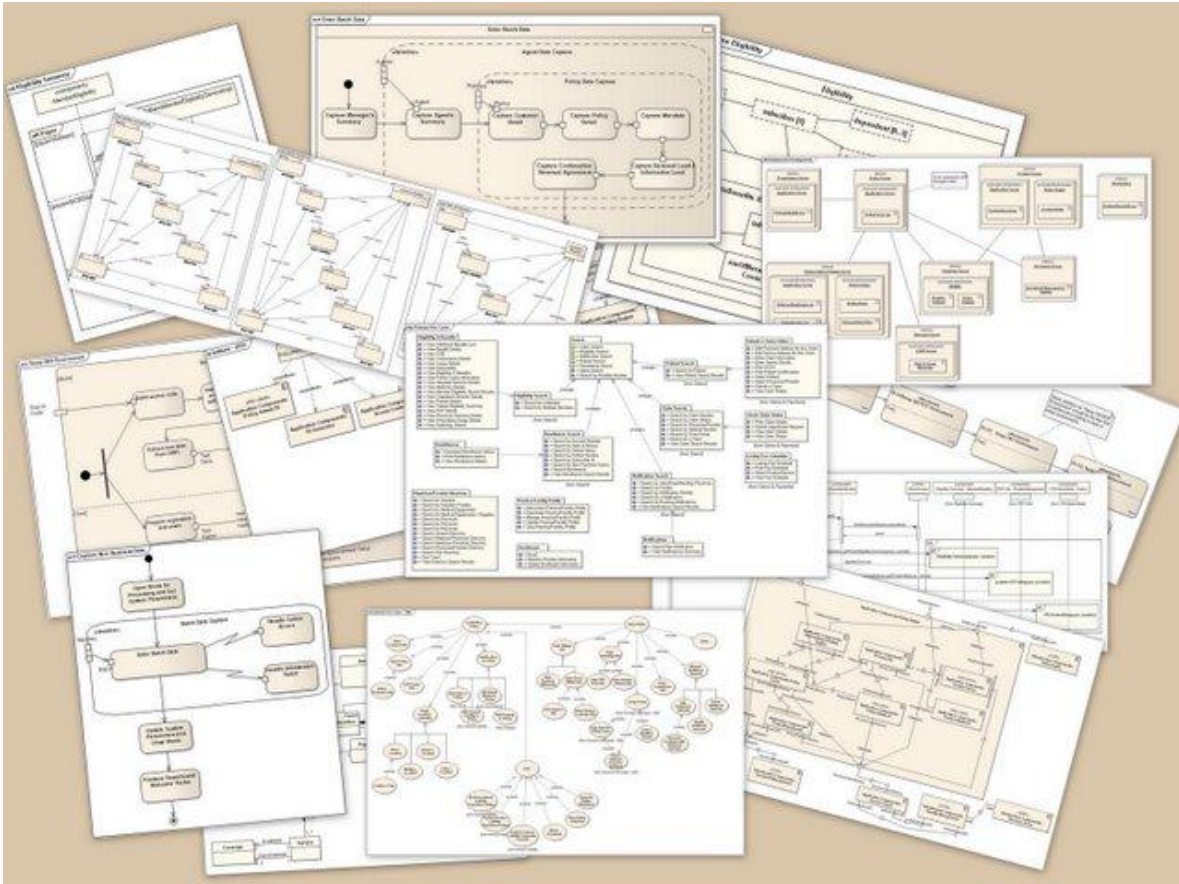


Figura #2 vistas de UML fuente: google

10.1. Diagramas

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, a continuación se mencionarán los que son parte de UML y posteriormente se describirán los que se utilizaron en el desarrollo del sistema.

- ❖ Diagramas de clases
- ❖ Diagramas de objetos



- ❖ Diagramas de casos de usos
- ❖ Diagramas de estados
- ❖ Diagramas de secuencias
- ❖ Diagramas de actividades
- ❖ Diagramas de colaboración
- ❖ Diagramas de componentes
- ❖ Diagramas de distribución

Diagrama de Clases

Los diagramas de clase muestran las características estáticas del sistema y no representan ningún procesamiento en particular, también destacan la naturaleza de las relaciones entre las clases.

Un diagrama de clase puede mostrar solo el nombre de la clase, el nombre de la clase y los atributos o el nombre de la clase, los atributos y los métodos. Es útil mostrar sólo el nombre de la clase cuando el diagrama es muy complejo e incluye muchas. (Kenneth E. Kendall, 2011)

En el párrafo anterior se mencionaron dos conceptos bien importantes que contiene una clase que son los atributos y métodos a continuación se definirán estos conceptos:

- ❖ **Atributos:** Es una propiedad o característica de lo que la clase sabe acerca del objeto.
- ❖ **Métodos (Operaciones):** Es una operación que la clase puede realizar, en pocas palabras los métodos no son más que rutinas de código que trabajan con los atributos.

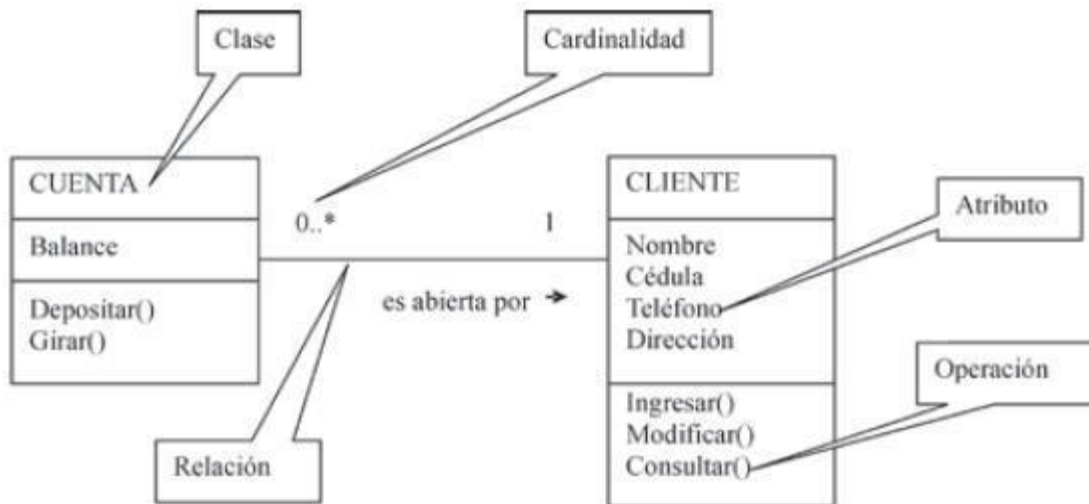


Figura #3 Diagrama de Clase. Fuente: google

Modelo de Casos de Usos

UML se basa fundamentalmente en una técnica de análisis orientado a objetos conocida como “modelado de casos de uso”, por lo cual describe que hace el sistema sin describir como lo hace. Podemos utilizar UML para analizar el modelo de casos de uso y derivar los objetos del sistema junto con sus interacciones entre sí y con los usuarios. (Kenneth E. Kendall, 2011)

El modelo de caso de uso proporciona medios eficaces de comunicación entre el equipo del negocio y el equipo de desarrollo. Un modelo de caso de uso divide la funcionalidad del sistema en comportamientos, servicios y respuestas que son significativos para los usuarios del sistema

En los diagramas de casos de usos existen dos elementos importantes, los casos de uso y los actores:

- ❖ **Casos de uso:** Larman lo describe como un documento narrativo que relata la secuencia de eventos de un actor, son historias o casos de utilización del sistema, no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican tácticamente los requerimientos en las historias que narran (Larman, 2003)



- ❖ **Actores:** Es una entidad externa del sistema que de alguna forma participa en los casos de usos, por lo general estos estimulan al sistema con eventos de entrada de datos al sistema o algo del actor (Larman, 2003)

Identificación de Casos de Usos

Un método con que se identifican los casos de usos se basa en los actores:

- ❖ Se identifican los actores relacionados con un sistema o empresa.
- ❖ En cada actor, se identifican los procesos que inician o en que participan.

Un segundo método de identificación de los casos de usos se basa en los eventos:

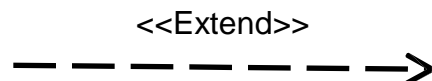
- ❖ Se identifican los eventos externos a los que un sistema ha de responder.
- ❖ Se relacionan los eventos con los actores y con los casos de uso.

Relaciones del Caso de Uso

Asociación: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.



Extensión: La inserción de comportamiento adicional en un caso de uso base que no tiene conocimiento sobre él.

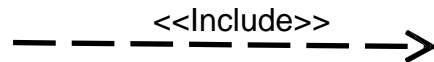


Generalización de casos de uso: Una relación en un caso de uso general y un caso de uso más específico, que hereda y añade propiedades a aquel.





Inclusión: Inserción de comportamiento adicional en un caso de uso base, que describe explícitamente la inserción.



Desarrollo de Diagramas de Casos de Usos

Al diagramar un caso de uso, empiece pidiendo a los usuarios que mencione todo lo que el sistema debe hacer para ellos. Escriba quien está involucrado con cada caso de uso y los servicios que el caso de uso debe proporcionar a los actores. Use los siguientes lineamientos (Kenneth E. Kendall, 2011)

- ❖ Revise las especificaciones del negocio e identifique los actores en el dominio del problema.
- ❖ Identifique los eventos de alto nivel y desarrolle los casos de uso principales que describen dichos eventos y como los inician los actores.
- ❖ Revise cada caso de uso principal para determinar las posibles variaciones del flujo a través del caso de uso. Debido a que el flujo de eventos es normalmente diferente en cada caso busque actividades que podrían tener éxito o fallar.

Caso de uso del almacén

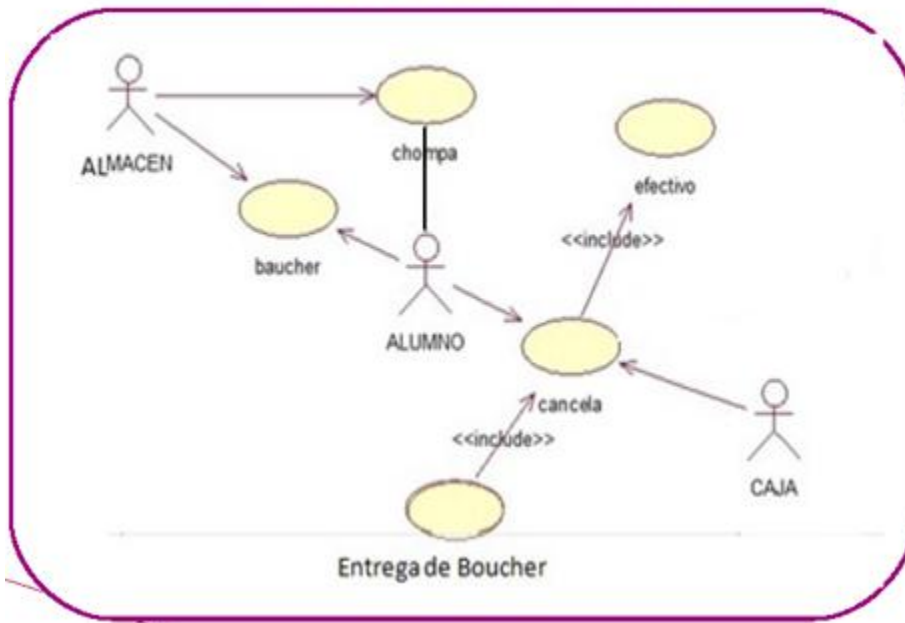


Figura #4 Diagrama de Caso de uso fuente:google

Desarrollo de Escenarios de Casos de Usos

Cada caso de uso tiene una descripción. Nos referimos a la descripción como un escenario de caso de uso. No hay ningún formato estándar de escenario de caso de uso, de modo que cada organización se enfrenta con especificar qué estándares se deben incluir. Un escenario de caso de uso se divide en tres secciones:

- ❖ Identificadores e iniciadores de caso de uso
- ❖ Pasos desempeñados
- ❖ Condiciones, suposiciones y preguntas.

Diagrama de secuencia

Kendall describe los diagramas de secuencias como una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Estos diagramas se derivan de los casos de usos y se utilizan para mostrar el patrón general de las actividades o interacciones en un caso de uso. (Kenneth E. Kendall, 2011)

En un diagrama de secuencia, un objeto se representa como una caja en la parte superior de una línea vertical punteada, esta línea vertical se llama línea de vida del objeto. Esta representa la vida del objeto durante la interacción.

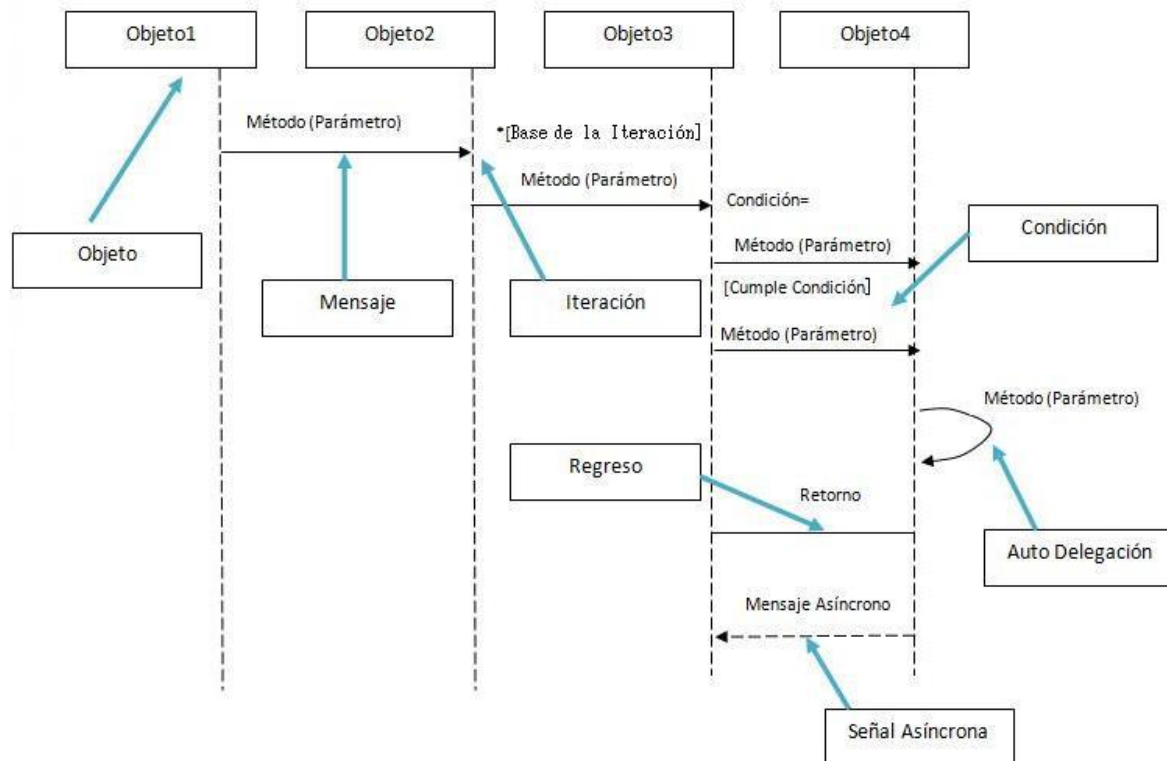


Figura #5 Diagrama de secuencia. Fuente:google

Cada mensaje se representa por medio de una flecha entre la línea de vida de dos objetos, el orden en el que transcurren los mensajes es de arriba hacia abajo, cada mensaje es etiquetado por lo menos con el nombre del mensaje, aunque también puede incluirse los argumentos o alguna información de control,



también se puede mostrar la auto delegación que es un mensaje de vuelta a la misma línea de vida (Fouler, 1999)

10.2. Diccionarios de Datos

Es una aplicación especializada de los tipos de diccionarios usados como referencia en la vida cotidiana. El diccionario de datos es una obra de consulta con la información acerca de los datos (es decir meta datos), compilado por los analistas de sistemas para guiarse en el análisis y diseño (Kenneth E. Kendall, 2011)

Elementos de Datos

Los siguientes son los elementos que comúnmente se incluyen en el formulario de descripción de elementos (Kenneth E. Kendall, 2011)

- ❖ ID del elemento
- ❖ Nombre del elemento
- ❖ Alias
- ❖ Descripción breve del elemento
- ❖ Si el elemento es base o derivado
- ❖ Longitud del elemento
- ❖ Tipo de datos
- ❖ Formato de entrada y salida
- ❖ Criterios de validación
- ❖ Cualquier valor predeterminado (Un área adicional para observaciones o comentarios.)

11. Universidad Nacional Agraria (UNA)

La UNA es una institución de enseñanza superior agrícola más antigua y de mayor experiencia en Nicaragua. Durante sus años de existencia ha trabajado en pro de una transformación sistemática, cumpliendo la misión de formar profesionales y



desarrollar programas de investigación y extensión agropecuaria, respondiendo a los diferentes modelos de desarrollo que el país ha experimentado.

El 25 de mayo de 1917 por Decreto Ejecutivo, se constituye la Escuela Nacional de Agricultura, en el Diario Oficial La Gaceta No. 157 que da a conocer el Reglamento Interior de la Escuela Nacional de Agricultura, mismo que estipula lo siguiente:

1. En la Escuela Nacional de Agricultura se estudiarán las asignaturas necesarias para obtener el Título de Labrador Científico, conforme a lo preceptuado en el decreto de creación del 25 de mayo último.
2. Para ingresar en la Escuela, habrá que atenerse en un todo a lo preceptuado en el citado decreto.
3. La enseñanza es completamente gratuita para los alumnos pensionados por el Gobierno. Los derechos de matrícula para todos los demás alumnos, serán los exigidos en los artículos 49 y 56 del mismo decreto. Así como los de examen y títulos correspondientes.
4. Los que no presentaren el certificado de aprobación de cuatro primeros años de enseñanza primaria, sufrirán un examen con arreglo a lo preceptuado en el artículo 54 del Reglamento General.

El 18 de marzo 1936, por decreto presidencial, se aprobó la Ley que reglamentó los estudios de Agricultura y Veterinaria, Ley publicada en la Gaceta No. 262, que en su Arto. 1, estipula: “Autorizar a la Secretaría de Estado en el Despacho de Instrucción Pública y Educación Física para que en lo sucesivo extienda los títulos de Perito Agrícola y Experto en Veterinaria a los alumnos de la Escuela Nacional de Agricultura de Chinandega” (La Gaceta, 1937).

El 14 de marzo de 1949 por Decreto Presidencial No. 23 oficializado en Las Gacetas No. 107 -117, se da a conocer el Reglamento General de la Escuela Nacional de Agricultura y Ganadería (ENAG), con el fin de preparar con conocimientos teóricos – prácticos a los jóvenes del país para ejercer la profesión agrícola y zootécnica. Los alumnos debían participar en los trabajos y ocupaciones



requeridas para la buena marcha de la Escuela; cuya organización tenía una concepción de explotación privada, en la que el objeto del cultivo o industria no perseguía otro fin que el de la enseñanza del alumno. La duración de los estudios era de cuatro años, y obtenían el título de Perito en Agronomía y Zootecnia.

Visión

Es una institución líder en Educación Superior Agraria, caracterizada por su calidad, eficiencia y transparencia, con impacto nacional y proyección regional e internacional en la formación de profesionales, en tanto contribuye con la generación de conocimientos científico-técnicos e innovación para el desarrollo agrario integral y sostenible.

Es reconocida por su vinculación e integración al desarrollo regional y nacional a través de programas académicos pertinentes, flexibles e innovadores que abarcan diferentes áreas del conocimiento agrario y son desarrollados en ambientes que fomentan el aprendizaje significativo, con escenarios variados y utilización de tecnologías de comunicación apropiadas para la construcción del conocimiento y el desarrollo de competencias técnicas y valores.

Es una institución consolidada orgánicamente, con una estructura flexible, dinámica y adaptada al cambio. Los miembros de la comunidad están comprometidos con la calidad en el desarrollo de todos los procesos y procedimientos académicos y administrativos.

Misión

La Universidad Nacional Agraria es una institución de educación superior pública, autónoma, sin fines de lucro, que contribuye desde la perspectiva del Compromiso Social Universitario, al desarrollo agrario integral y sostenible, y a la conservación del ambiente, mediante la formación de profesionales competentes con valores éticos, morales y cultura ambientalista; la construcción de conocimiento científico y tecnológico y la producción, gestión y difusión de información.

11.1. Requisitos de Admisión



EL Consejo Universitario de la Universidad Nacional Agraria (UNA) en uso de las atribuciones que le concede el Arto. 16, numeral 13 de la Ley de Autonomía de las Instituciones de la Educación Superior que textualmente dice: “Aprobar el calendario académico y las políticas de ingreso”, presenta la Normativa de Admisión para sus diferentes modalidades de estudio y sedes regionales:

1. Metódica para el primer ingreso a la modalidad presencial.

El proceso de admisión para el primer ingreso modalidad presencial a la UNA es único y obligatorio para todos y cada uno de los aspirantes que cumplan con los requisitos académicos establecidos, consta de las siguientes fases:

Pre-Matrícula

La pre-matrícula en modalidad presencial en la UNA, tiene como propósito la inclusión de los aspirantes en el listado oficial, que les da derecho a realizar el curso propedéutico. Esta se realiza en caja y registro central o sede regional correspondiente.

Los aspirantes pueden realizar la pre-matrícula en cualquiera de las siguientes situaciones:

- ❖ Los estudiantes que cursan su último año de educación secundaria.
- ❖ Los bachilleres que nunca se han matriculado en la UNA.
- ❖ Los aspirantes que fueron estudiantes de primer año de la UNA que oficialmente hallan retirado su matrícula.
- ❖ Los estudiantes de otras universidades que deseen continuar sus estudios en la UNA.
- ❖ Los bachilleres graduados en el extranjero con su diploma debidamente certificado.



11.2. Requisitos para realizar pre-matrícula:

- ❖ Certificado original de notas de IV año y I semestre de V año, extendido en su centro de estudios para los que cursan el V año de bachillerato.
- ❖ Para los graduados, su diploma de bachiller
- ❖ 2 fotografías tamaño carne.
- ❖ Partida de nacimiento o cedula de identidad.
- ❖ Pago del arancel.

Realización de un Curso Propedéutico para los aspirantes a la modalidad regular o presencial

Las actividades del curso propedéutico permitirán la ubicación de los estudiantes en las diferentes carreras que oferta la UNA de acuerdo a sus capacidades y cupos disponibles en la universidad. El curso tiene como propósito:

- ❖ Mejorar las habilidades cognitivo-lingüísticas, analíticas y numéricas de los aspirantes para permitirles su inserción optima al ambiente de aprendizaje universitario.
- ❖ Desarrollar actitudes, valores y hábitos necesarios en la vida universitaria.
- ❖ Brindar información suficiente y necesaria a los aspirantes para facilitarles su ingreso, de acuerdo con su vocación, aspiración, interés, y oferta académica de la UNA.

Ubicación en la Carrera por parte de la UNA

El proceso de ubicación se fundamenta en los criterios siguientes:

Los cupos previamente establecidos por carrera.

Los promedios (de mayor a menor) obtenidos por los aspirantes tomando en consideración el promedio de secundaria y los resultados del curso propedéutico.

La opción (primera y segunda) de los aspirantes.



Matrícula

La matrícula es el registro que se le hace al aspirante por primera vez en la UNA, esta actividad se realiza de forma personal.

2. Metodica para el primer ingreso a la modalidad a distancia

El proceso de admisión para el primer ingreso modalidad a distancia a la UNA es único para todos y cada uno de los aspirantes que cumplan con los requisitos académicos establecidos, consta de las siguientes fases:

Pre-Matrícula

La pre-matrícula en modalidad a distancia en la UNA, tiene como propósito la inclusión de los aspirantes en el listado oficial que les da derecho a un cupo en la modalidad. Esta se realiza en caja y registro central o sede regional correspondiente.

Los aspirantes pueden realizar la pre-matrícula en cualquiera de las siguientes situaciones:

- ❖ Los estudiantes que cursan su último año de educación secundaria.
- ❖ Los bachilleres que nunca se han matriculado en la UNA.
- ❖ Los aspirantes que fueron estudiantes de la UNA y deseen continuar sus estudios.
- ❖ Los estudiantes de las universidades que deseen continuar sus estudios en la UNA.
- ❖ Los bachilleres graduados en el extranjero con su diploma debidamente certificado.
- ❖ Técnicos superiores.
- ❖ Profesionales que deseen estudiar una segunda carrera.



Requisitos para realizar pre-matrícula:

- ❖ Certificado original de notas de IV año y I Semestre de V año, extendido en su centro de estudios, para los que cursan el V año de bachillerato.
- ❖ Para los graduados, su Diploma de bachiller o técnico superior...
- ❖ 2 fotografías tamaño carné.
- ❖ Partida de nacimiento o cédula de identidad
- ❖ Pago del arancel

11.3. Sistema de Evaluación

Artículo 84: El sistema de evaluación del aprendizaje constituye un elemento del proceso educativo, cuya finalidad es determinar el grado de cumplimiento de los objetivos y/o logros alcanzados en las asignaturas, módulos prácticos y practicas pre-profesionales, para la autorregulación de los aprendizajes; así como reorientar las actividades de dicho proceso en casos necesarios. El sistema de evaluación de la UNA comprende:

- a) Evaluación sistemática
- b) Evaluaciones parciales
- c) Evaluación cualitativa
- d) Evaluación extraordinaria
- e) Evaluación por suficiencia

Artículo 85: En la primera sesión de clases, los docentes darán a conocer a los estudiantes el sílabo, la metodología de enseñanza, bibliografía y las formas de evaluación establecida en el programa de la asignatura.

Artículo 86: La escala de calificación para la evaluación de las asignaturas en la UNA es de 0 a 100, considerándose:



0 a 59 Deficiente

60 a 69 Regular

70 a 79 Bueno

80 a 89 Muy Bueno

90 a 100 Excelente

Para todas las formas de evaluación, se considera:

60 a 100 Aprobado

0 a 59 Reprobado

Artículo 87: La evaluación sistemática consiste en la realización de preguntas de control y de comprobación, realización de seminarios, clases prácticas, laboratorios, prácticas de campo, pruebas cortas y trabajos independientes. Este tipo de evaluación no excede el 40% de la nota parcial respectiva.

Artículo 88: En la evaluación parcial el estudiante acumula dos calificaciones a lo largo de cada semestre o cuatrimestre. Una a mitad del semestre o cuatrimestre y otra al final del período.

Artículo 89: Cada evaluación parcial comprende la sumatoria de las evaluaciones sistemáticas y la del examen parcial correspondiente. El porcentaje a otorgarse al examen parcial es el 60% de la nota parcial. Al estudiante que no realice uno de los exámenes parciales de manera injustificada. La calificación parcial estará constituida por el porcentaje acumulado en las evaluaciones sistemáticas.

Artículo 90: En los planes de estudio donde se contemplan asignaturas con trabajos de curso, se les otorga un porcentaje del 60% del valor de la segunda evaluación parcial.

Artículo 91: La calificación final de cada asignatura se obtiene mediante el promedio de las calificaciones parciales.



Artículo 92: El contenido de las pruebas correspondientes a las distintas formas de evaluación exceptuando las sistemáticas, será revisado y aprobado por el Jefe del Departamento Académico, quien podrá delegar esta tarea en el profesor de mayor experiencia en la asignatura.

Artículo 93: Los exámenes parciales, extraordinarios y por suficiencia serán por escrito y tendrán una duración máxima de 100 minutos. Las pruebas escritas de corta duración se desarrollarán en un tiempo máximo de 20 minutos y las orales un máximo de 10.

Artículo 94: El estudiante tiene derecho a recibir de parte del docente los exámenes y trabajos corregidos y brindar consultas relativas a éstos en concordancia con lo establecido en el **Arto 54** del Reglamento del Trabajo Docente.

Artículo 95: En caso que el estudiante considere que no ha sido evaluado con los parámetros establecidos en la evaluación, tiene derecho a apelar ante el Jefe del Departamento Académico o Director de Sede que sirve la asignatura quien analiza la petición, si la encuentra justificada designará a dos profesores titulares del área del conocimiento o al docente de mayor experiencia, para que efectúen la revisión. Si se demuestra que la calificación otorgada por el profesor no es correcta, la nota definitiva correspondiente a la evaluación reclamada, será la calificación asignada por el revisor o revisores, en caso contrario se ratifica la nota original.

Artículo 96: El estudiante tiene derecho a conocer los resultados de sus evaluaciones en un término máximo de tres días hábiles después de aplicada la evaluación, en caso de asignaturas que culminen con trabajo de curso, un máximo de cinco días hábiles después de la fecha de calendarización de la segunda evaluación parcial correspondiente a la asignatura.



Artículo 97: El estudiante puede ingresar al aula de clase, en los primeros quince minutos de iniciada la actividad académica, sin que esta se le contabilice como inasistencia; transcurrido este tiempo queda ausente para la primera hora, pudiendo integrarse a su actividad académica después del receso de los primeros cincuenta minutos de clase con derecho a su asistencia.

Artículo 98: El estudiante que no se presente a cualquier tipo de evaluación será calificado con la nota de NSP, exceptuando los casos justificados en tiempo y forma para su respectiva programación.

Artículo 99: El estudiante que incurra en inasistencia a cualquier tipo de evaluación, podrá justificarla ante el docente dentro de los dos días hábiles de ocurrida ésta. En el caso que se exceda en tiempo, el estudiante deberá presentar la debida justificación por escrito ante el Secretario Facultativo o Académico de sede, en un plazo no mayor de cinco días hábiles de la inasistencia, quien podrá denegar o aprobar la reprogramación de la evaluación tomando en consideración las causas siguientes, a excepción de las actividades de formación práctica:

- a) Embarazo.
- b) Misión Universitaria (edecanes, cultura, deporte, monitores, jornadas, divulgación).
- c) Accidente grave.
- d) Enfermedad con indicación de reposo médico.
- e) Muerte de un familiar hasta tercer grado de consanguinidad y primer grado de afinidad.
- f) Misión de trabajo laboral previamente autorizada por la universidad.
- g) Misiones oficiales del gremio estudiantil, presentando constancia de la actividad.
- h) Permisos especiales autorizados con anticipación.



Artículo 100: Para poder presentarse al segundo examen parcial el estudiante deberá haber asistido al menos a un 80% del total de horas impartidas. Exceptuando las causales del Art. 100 permitiendo un mínimo del 60% de la asistencia.

Artículo 101: Toda reprogramación de exámenes contempla el pago previo de un arancel establecido por el Consejo Universitario, excepto a los cubiertos por el Art. 100, en los incisos b, g y h.

Artículo 102: El estudiante reprobado cuya nota final sea igual o menor a 59 puntos, tiene derecho a realizar un examen extraordinario en las asignaturas inscritas del período que cursa, siendo requisito haber realizado los dos exámenes parciales. En dicho examen se evalúa el contenido programático de la asignatura cursada en el semestre o cuatrimestre. El estudiante se presenta sin promedio de entrada y si aprobare, su nota final será de 60 puntos.

Artículo 103: En la modalidad presencial el curso de verano se evalúa con dos calificaciones parciales con valor del 50% cada una.

Artículo 104: La inscripción a los cursos de verano, se realizará en las Secretarías Facultativas o Secretaría Académica de Sede. La cantidad de estudiantes para iniciar un curso de verano no debe ser menor de 5 estudiantes. Un grupo de curso de verano no podrá ser mayor de 30 estudiantes.

Artículo 105: El arancel de los cursos de verano es aprobado por el Consejo Universitario y asumido por los estudiantes inscritos en el curso.

Artículo 106: El estudiante que considere tener dominio de una asignatura por su experiencia profesional o por la obtención de otros cursos relacionados, podrá solicitar por escrito autorización para la realización de un examen por suficiencia.



Artículo 107: El estudiante tiene derecho a presentar como máximo cinco exámenes por suficiencia a lo largo de su vida académica una vez en cada asignatura autorizada; siempre que no haya cursado y aplazado previamente la asignatura a evaluar.

Artículo 108: Para realizar el examen por suficiencia el estudiante debe presentar su solicitud ante la Secretaría Facultativa o Secretaría Académica de Sede, un mes antes del semestre o cuatrimestre académico correspondiente a la ubicación de la asignatura. La Secretaría correspondiente responderá a las solicitudes presentadas a más tardar 10 días hábiles después de recepcionada.

Artículo 109: El estudiante junto con su autorización deberá inscribir la asignatura, siendo contabilizada en su carga académica en el período correspondiente y recibir fotocopia del programa a examinar, previo pago del arancel correspondiente aprobado por el Consejo Universitario. La calificación obtenida será incorporada al expediente académico del estudiante y equivaldrá al resultado del curso regular o por encuentro respectivo.

Artículo 110: El estudiante que resultare reprobado en el examen por suficiencia, queda obligado a inscribir y cursar la asignatura de manera ordinaria, considerándose ésta como la segunda vez que la cursa.

Artículo 111: En los casos de no existir el número mínimo de estudiantes para un curso de verano, podrá recurrirse al curso vía tutoría, el cual considera un mínimo de 20 horas de contacto docente-estudiante. El curso de tutoría se evalúa con dos calificaciones parciales con valor del 50% cada una, el arancel de tutoría será aprobado previamente por el Consejo Universitario.

Artículo 112: El curso vía tutoría se aplicará fundamentalmente en proceso de recuperación de asignaturas reprobadas.



Artículo 113: Los módulos prácticos se evalúan mediante el sistema de evaluación cualitativa, teniendo como base la asistencia, disciplina, habilidades alcanzadas y el reporte final del módulo, conteniendo todas las tareas y/o actividades realizadas por el estudiante durante el desarrollo del mismo. El reporte final deberá ser entregado por el estudiante al docente en la fecha estipulada durante el período que comprende la segunda evaluación parcial.

Artículo 114: Las prácticas pre-profesionales, los ejercicios profesionales supervisados y pasantías son evaluadas por el docente supervisor de las prácticas, teniendo en cuenta: La calidad del informe del estudiante, la evaluación de la instancia donde realizó la práctica y la valoración del docente supervisor, iniciativa e independencia demostrada por el estudiante en la realización de la misma.

Artículo 115: Los módulos prácticos, las Prácticas Pre-profesionales, los Ejercicios Profesionales Supervisados y las pasantías se evaluarán mediante la siguiente escala de calificación:

Valoración cualitativa	Equivalente Numérico
D: Deficiente o reprobado	Menor 70
B: Bueno	70-79
MB: Muy Bueno	80-89
E: Excelente	90-100

Las categorías “E”, “MB” y “B” son de aprobado. La categoría “D” es de reprobado, deberán constar en el acta de la calificación respectiva al finalizar la práctica. El equivalente numérico es para uso referencial para certificados emitidos por la Oficina de Registro Académico.



Artículo 116: Para poder aprobar los módulos prácticos el estudiante deberá haber asistido al 80% de las horas prácticas. En el caso de las prácticas pre-profesionales, los ejercicios profesionales supervisados y las pasantías deberán de haber asistido al 90% de los días programados para la realización de las prácticas.

Artículo 117: El estudiante que obtenga la calificación final “D” (deficiente o reprobado) en los módulos prácticos, las prácticas pre-profesionales, los ejercicios profesionales supervisados y las pasantías deberá de repetirla, sujetándose a los procedimientos de las asignaturas reprobadas (UNA)

V. Diseño Metodológico

1. Tipo de investigación a realizar

La investigación que se efectuó fue de tipo aplicada, cuya realización pretenden resolver problemas prácticos del mundo moderno, el objetivo de la investigación aplicada es mejorar la condición humana.

Para su realización se tomaron como base un conjunto de conocimientos teóricos y prácticos adquiridos durante estos años de estudio y aprendizaje. Por el tiempo es de tipo transversal.

2. Universo y muestra

2.1. Universo

Las Universidades.

Nuestra aplicación se puede implementar en otras Universidades. Donde los estudiantes puedan sacarle provecho al avance de la tecnología y con esta aplicación facilitar sus gestiones académicas con solo contar con un teléfono con sistema operativo android y conexión a internet o paquetes de datos móviles.



2.2. Muestra

Universidad Nacional Agraria (UNA)

Es la muestra en donde se desarrolló la aplicación móvil, lo cual ha mostrado mucho interés por los estudiantes más por los que no poseen acceso a una computadora.

3. Obtención de los datos

La recopilación de la información necesaria para desarrollar esta investigación, se realizó a través de la observación, entrevistas al responsable de la administración del registro académico y encuesta a estudiantes de la Universidad Nacional Agraria.

3.1. La observación

La observación fue uno de los instrumentos de medición que mejor se adaptado para el proceso de análisis. Esta técnica se implementó mediante visitas realizada a la Universidad Nacional Agraria (UNA), donde determinamos las condiciones tecnológicas que posee, procesamiento de datos vital para la aplicación.

3.2. La entrevista

En el desarrollo de la aplicación se realizaron entrevistas abiertas, estas nos permitió conocer más objetivamente del procesamiento de la información académica de los estudiantes y cuáles son sus requerimientos necesarios para cada módulo de la aplicación.



3.3. La encuesta

La encuesta fue una de las herramientas primordiales en el desarrollo de esta investigación la cual nos ayudó a estudiar una población mediante el análisis de muestras representativas a fin de explicar las variables de estudio y su frecuencia.

De acuerdo a la forma de obtención de la información que obtuvimos, las encuestas las clasificamos en:

- ❖ Entrevistas
- ❖ Cuestionarios por correo
- ❖ Panel
- ❖ Entrevistas por teléfono

En el desarrollo de la investigación se realizaron encuestas; estas encuestas tienen como finalidad determinar los sistemas operativos móviles que utilizan los estudiantes y obtener información objetiva acerca del acceso al realizar una consulta.

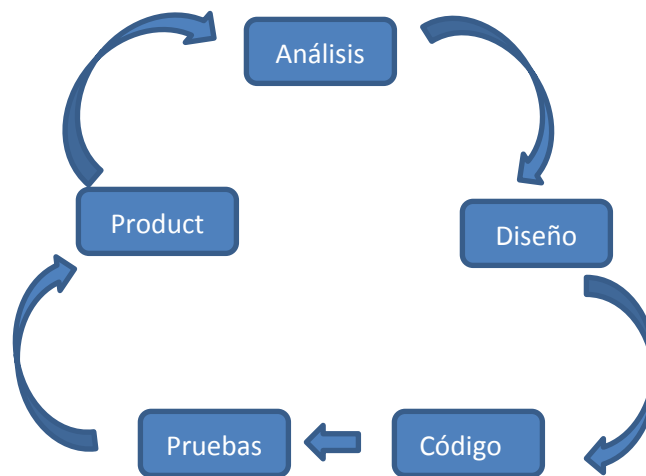
4. Metodología utilizada para la producción de software

El proceso de creación del software (Aplicación móvil) se rige por los lineamientos del **Modelo Incremental**, el cual es uno de los paradigmas más eficiente de la ingeniería del software.

- ❖ Usamos este modelo ya que el cliente no tiene que esperar el producto final para ser aprovechado, cada incremento satisface los requerimientos de tal forma que puede utilizar la aplicación inmediatamente.
- ❖ Existe bajo riesgo, porque a través de este método se depuran los posibles que se puedan presentar en cada uno de los incrementos.

- ❖ Cada agregado puede ser usado para explorar el prototipo y obtener experiencia para los requerimientos de los incrementos posteriores de la aplicación.

5. Fases del modelo incremental



5.1. Fase en el desarrollo del sistema

A lo largo del avance de la aplicación Móvil, esta ha atravesado por varias etapas obteniendo diferentes resultados los cuales han fortalecido la eficacia de la aplicación. Durante el desarrollo de esta aplicación se logró realizar dos incrementos.

5.1.1. Incremento 1

En el primer incremento se abordó el módulo de consultas de calificaciones en lo que se obtuvieron las siguientes fases:

Fase de Análisis

En esta fase se definió el modelo del negocio y los alcances del proyecto. Para lograr esto, se realizaron varias visitas a la organización con la



finalidad de determinar las condiciones tecnológicas presente en la Universidad Nacional Agraria (UNA).

En esta etapa se obtuvieron los siguientes resultados:

- ❖ Se estableció la visión general de los requerimientos del proyecto.
- ❖ Se nos brindó información necesaria y oportuna.
- ❖ Planteamiento de los alcances a lograr con la aplicación.
- ❖ Se definió la arquitectura para la aplicación.
- ❖ Se mostró interés durante el desarrollo de la aplicación.

Fase de diseño

En esta fase se analizó el dominio del problema y se desarrolló el plan del proyecto.

Se obtuvieron los siguientes resultados:

- ❖ Se elaboró un modelo de caso de uso completo
- ❖ Se hizo una reingeniería a la base de datos existente mejorando la integridad de los datos.
- ❖ Se realizó la normalización y creación de la base de datos (mysql).
- ❖ Se desarrolló una descripción de la arquitectura de la aplicación

Fase de código

Consistió en el proceso de programación de la aplicación móvil y demás módulos de la misma.

Logros de esta etapa:

- ❖ Creación del diseño físico de las base de datos
- ❖ Prototipos operacionales de los módulos de consultas.
- ❖ Diseño físico de las base de datos.
- ❖ Realización de pruebas necesarias en cada uno de los módulos que se



desarrolló.

- ❖ Se hicieron pruebas de accesibilidad de la aplicación en un servidor remoto temporal, con las versiones de los dispositivos que se dispone para la realización de pruebas.(2.2,2.3,4.2)

5.1.2. Incremento 2

En el segundo incremento abordamos el módulo de resultados de admisión obteniendo las siguientes fases:

Fase de Análisis

En esta etapa se obtuvieron los siguientes resultados:

- ❖ Se estableció la visión general del requerimiento para el módulo.
- ❖ Se nos brindó información necesaria y oportuna.
- ❖ Planteamiento de los alcances a lograr con el módulo de resultados de admisión.
- ❖ Se definió la arquitectura para el módulo de la aplicación.

Fase de diseño

En esta fase se analizó el dominio del problema y se desarrolló el plan del proyecto.

Se obtuvieron los siguientes resultados:

- ❖ Se elaboró un modelo de caso de uso completo
- ❖ Se hizo una reingeniería a la base de datos existente mejorando la integridad de los datos.
- ❖ Se realizó la normalización y creación de la base de datos (mysql).
- ❖ Se desarrolló una descripción de la arquitectura de la aplicación



Fase de código

Consistió en el proceso de programación del módulo de Resultado de Admisión.

Logros de esta etapa:

- ❖ Creación del diseño físico de las base de datos
- ❖ Prototipos operacionales del módulo de Resultado de Admisión.
- ❖ Diseño físico de la base de datos.
- ❖ Realización de pruebas necesarias.
- ❖ Se hicieron pruebas de accesibilidad de este módulo de la aplicación en un servidor remoto temporal con las versiones de los dispositivos que se dispone para pruebas.(2.2,2.3,4.2)

Fase de Prueba

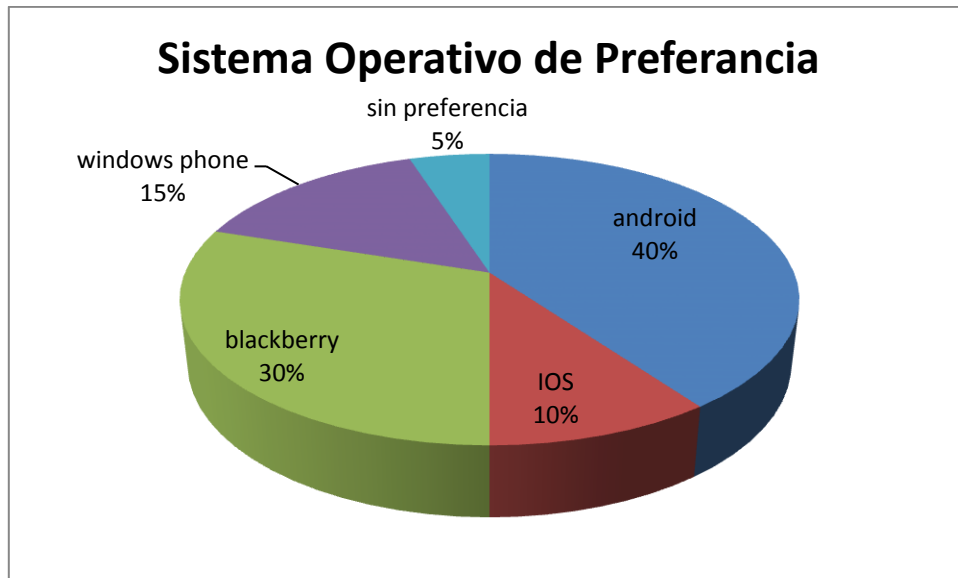
La fase de prueba es la última etapa del desarrollo de la aplicación móvil previo a la disposición de los usuarios finales a través del alojamiento de la aplicación en un servidor web accesible las 24 horas.

En esta fase se encuentran los dos módulos de la aplicación operacional y con accesibilidad, De tal manera que al finalizar esta fase se obtendrá un producto final que cumpla los requisitos esperados, asegurándose que todo funcione correctamente.

VI. Análisis de Resultado

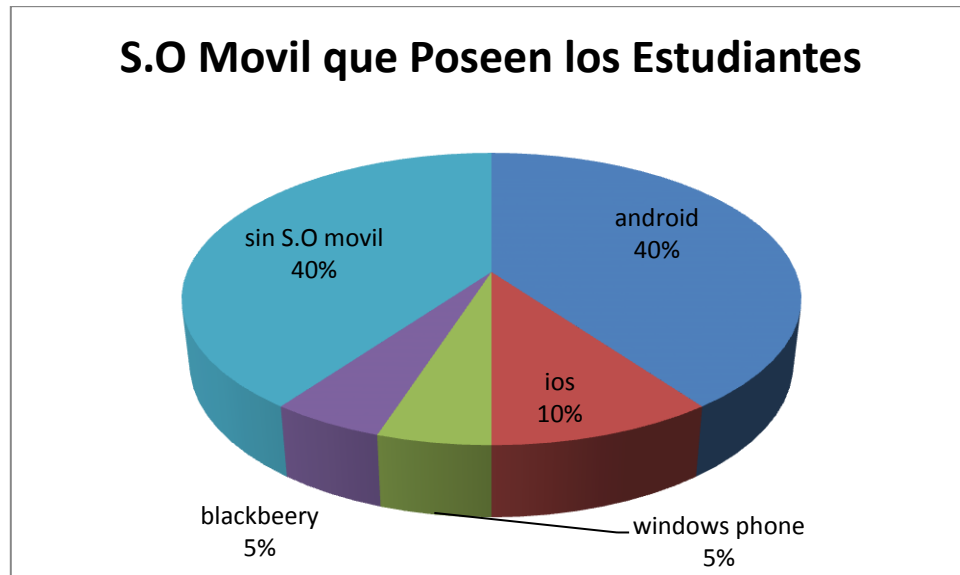
1. Encuesta

De los estudiantes encuestados obtuvimos que el sistema operativo Android posea una preferencia de 40%, IOS 10%, Windows Phone 15%, BlackBerry 30% y el 5% sin preferencia.

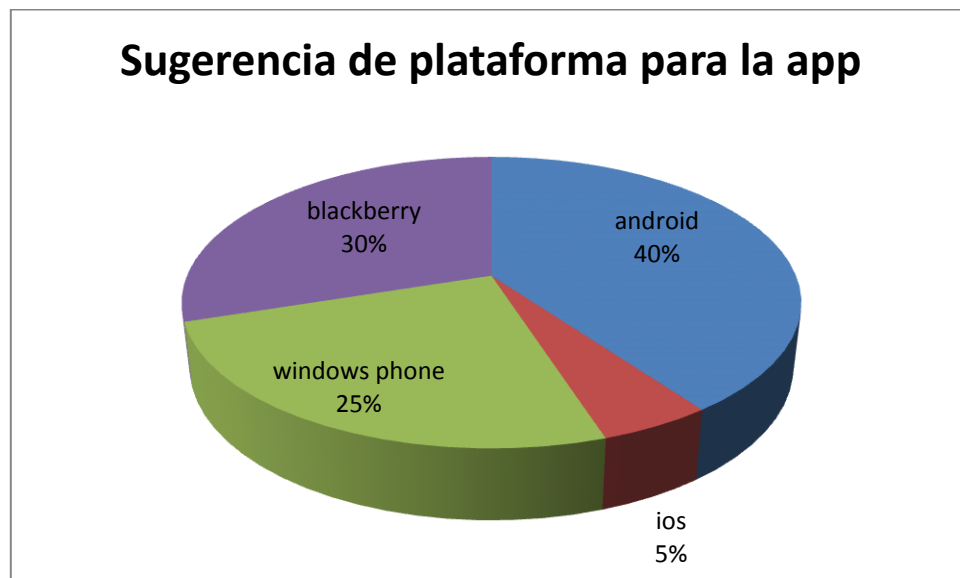




En la encuesta realiza se obtuvo que el 40% de los estudiantes poseen un teléfono móvil con sistemas operativos Android, IOS 10%, Windows Phone 5%, BlackBerry 5% y un 40% sin sistema operativo móvil.



Los estudiantes sugirieron varias plataformas de desarrollo para la aplicación siendo el sistema operativo Android con un 40% de aceptación, a diferencia de BlackBerry con 30%, Windows Phone 25% e IOS con 5%.





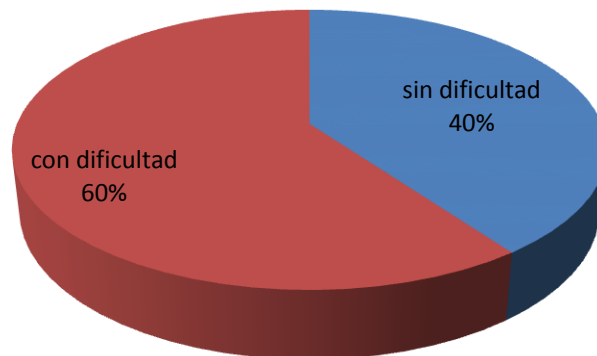
¿Qué información de consulta te resultaría más relevante en una aplicación móvil?





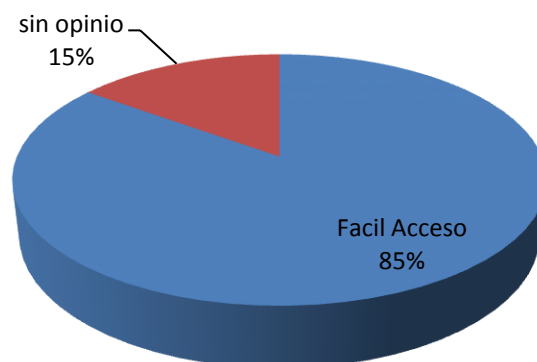
Los resultados de las encuestas realizadas a los estudiantes nos revela que un 40% de ellos no se les dificulta consultar información académica, a diferencia de otros; esta limitante al acceso de información se debe a muchos factores (no poseen computadora, no poseen en su móvil una aplicación que les permita al acceso a la información académica)

Acceso a informacion Academica



El 85% de los estudiantes testifican que si su móvil tuviese una aplicación Android se les facilitaría consultar la información académica desde su dispositivo móvil y el 15% de los estudiantes no opinaron.

Consulta de la Informacion Academica con app





2. Herramientas y Tecnologías

Se estableció la utilización de las siguientes herramientas y equipos de desarrollo para la creación del producto final.

Herramientas	Descripción
PhotoShop CSS6	Herramienta de diseño Grafico
MySQL	Gestor de base de datos
Embarcadero E-R	Herramienta de Modelado de Datos
Ksoap	Librería de servicio web en android
Soap	Librería protocolo de comunicación de datos asp.net
Visual studio 2010	Entorno de desarrollo integrado
Eclipse	Entorno de desarrollo
Microsoft Office 2010	Elaboración de documentación
Motorola XT316	Móvil para realización de pruebas
Motorola Defy	Móvil para realización de pruebas
LG E612	Móvil para realización de pruebas
Android development tools (ADT)	Extensión para desarrollo de Android bajo la plataforma eclipse

3. Descripción de la Aplicación

La aplicación fue perfeccionada bajo el entorno de desarrollo Eclipse para Android, se encuentra en la versión 4.4.2 y permite compatibilidad hasta la versión 2.2 para que se pueda ejecutar en la mayoría de los Smartphone con sistema operativo Android de la actualidad, esta compatibilidad es permitida gracias a Eclipse que ofrece esta opción, agregando al proyecto de la aplicación un proyecto llamado “Appcompat7”.

La conexión de datos que se utiliza se realiza vía servicio web, que permite obtener datos, manteniendo la privacidad de la información de la base de datos, el servicio web se encarga de solicitar la información al servidor de base de datos, de tal forma que el cliente (la aplicación), solo ve los datos que el servicio le permite.



La aplicación actualmente permite realizar dos consultas principales de carácter Académico, las cuales son:

3.1. Consulta de Resultados de Notas Semestrales

Esta información es de gran utilidad para los estudiantes activos de la Universidad Nacional Agraria ya que permitirá que la información de notas se consulte más fácilmente sin la necesidad de buscar esa información en murales, sitio web o teniendo la necesidad de contactar con el docente directamente.

3.2. Consulta de resultados de Admisión

Con esta información se pretende que los estudiantes de nuevo ingreso se vean beneficiados, de tal forma que puedan consultar si aprobaron el curso propedéutico, en qué carrera y su nota total, además de que les permitirá poder acceder a esa información en el lugar que se encuentren, ya que la mayoría son de departamentos fuera de Managua, para que no se vean en la necesidad de viajar para conocer dicho resultado.

Beneficios

- ❖ El acceso a la información de gran relevancia en poco tiempo.
- ❖ Facilidad operacional de la aplicación.
- ❖ El usuario cuenta con disponibilidad de la información a cualquier hora y en cualquier lugar donde cuente con conexión de datos.

4. Estudio de Factibilidad

Al inicio del proyecto se analizó los requerimientos técnicos (equipos a utilizar), económico (precio de los equipos) y la operatividad.

4.1. Factibilidad Técnica

En este momento la Universidad Nacional Agraria (UNA), cuenta con los equipos tecnológicos para alojar el servicio que brinda la aplicación móvil; a continuación se detallan los equipos que se utilizaron para la realización de este proyecto:



Cantidad	Recursos	Características
2	Servidor de Base de Datos	Intel Xeon E5-2403 1.8 GHZ ,4 núcleos disco memoria Ram 8 GB DDR3 1600 MHz disco duro 2 TB 7200 RPM DVD-RW 2 fuentes de poder Monitor 19 pulgadas
1	Laptop SAMSUNG NP270E4E	Capacidad de Almacenamiento 500 GB Intel CORE I3 cpu 3120M 2.5GHZ Procesador Gráfico Intel HD Graphics 400 Memoria Ram DDR3 4GB(expandible) Sistema Operativo Windows 7 ultimate unidad optica:CD+/-R,CD-RW,DVD-RW Web Cam incorporada(HD 720 P) Pantalla LED de 14"
1	Laptop LENOVO E531	Intel Core I3-2348M 2.5GHZ,3MB Cache Sistema Operativo Windows 7 Profesional Procesador Gráfico Intel HD 400 capacidad de Almacenamiento 1TB 5400rpm RAM hasta 16 GB Max DDR3(2 ranuras DIMM) Pantalla anti resplandor 15.6" HD web Cam de seguimiento de rostro(720p HD)
1	Laptop HACER ASPIRE ONE 722	disco duro 250GB HDD Sistema Operativo windows 7 HOME Premium Procesador AMD C-60 con Radeon™ 1.33 GHZ RAM 4GB DDR3 Pantalla LCD 11" HD ,1366 x 768 pixeles Procesador Gráfico ATI Radeon HD 6290 Graphcs
3	Gestor manejador de base de datos	My sql 6.8.3
3	plataforma desarrollo	Eclipse
1	Motorola XT316	Redes: claro o movistar



		Display:pantalla táctil 2.8",240x320 pixeles Conectividad: 3G(compatible(claro y mov) Sistema operativo:Android OS v2.3 (gingerbread)
1	LG-E612	Redes: claro o movistar Display:pantalla táctil 4",320x480 pixeles Conectividad: 3G(compatible(claro y mov) procesador de 800MHz Sistema operativo:Android 4.0 Ice Cream Sándwich
1	Motorola Defy	Redes: claro o movistar Display:TFT touchscreen, 16M colores 480 x 854 pixeles, 3.7" Conectividad: 3G(compatible(claro y mov) Sistema operativo:Android OS, v2.1 (Eclair)
3	plataforma desarrollo	Visual estudio 2010

4.2. Factibilidad Económica

En la parte económica se ha estimado lo primordial para desarrollo de la aplicación.

Descripción	Cantidad	costo unitario \$	Sub Total \$
Servidores	2	2022.85	4045.7
Laptop SAMSUNG	1	621.75	621.75
Laptop LENOVO	1	750	750
Laptop ACER ASPIRE ONE	1	300	300
Pago desarrolladores	3	800	2400
Eclipse	3	sin costo	sin costo
Visual Studio Ultimate 2010	3	200	600
My sql 6.8.3	3	sin costo	sin costo
LG-E612	1	160	160
Motorola defy	1	127	127
Motorola XT316	1	120	120



Papelería	2	4	12
Trasporte	4	7	21
Monto Total			9124.45

4.3. Factibilidad operacional

La factibilidad operativa determina la puesta en marcha de la aplicación.

Tanto los estudiantes que realizaron pruebas, como los encargados de garantizar la aprobación de la aplicación en la Universidad Nacional Agraria, consideran que la misma cumple con los requisitos y necesidades que están destinados a solucionar, ya que la aplicación es fácil de operar y muestra la información con el detalle necesario, además que la consulta se realiza en un tiempo óptimo y en cual lugar.



5. Diagrama UML

5.1. Casos de usos

Diagrama Estudiante Activo UNA



Diagrama Aspirante UNA





5.2. Descripción de Casos de Usos

Nombre del caso de uso: consultas de nota	Id único: carnet 04010231
Actor: Estudiante	
Descripción: permite al estudiante realizar consultas de calificaciones	
Evento desencadenador: El estudiante usa la App consultas sus calificaciones, introduce carnet y pin de estudiante, hace clic en consultar.	
Pasos a realizar	Información para los pasos
1. El estudiante inicia App	1.Carnet estudiante, Pin
2. La App muestra el Historial académico	2.Registro de Calificaciones
3. El actor elige guardar o salir	3.Botón(guardar/salir)
Pre-condiciones: El estudiante se encuentre en el registro académico	
Post-condiciones: El estudiante consulto con éxito su historial académico	
Suposiciones: El estudiante tiene un Smartphone con conexión web, carnet y pin	

Nombre del caso de uso: Resultado Admisión	Id único: pre matrícula 0715
Actor: Aspirante	
Descripción: permite al Aspirante ver resultado de admisión	



Evento desencadenador: El estudiante usa la App consulta resultado, Introduce n° pre matrícula, hace clic en consultar.	
Pasos a realizar	Información para los pasos
1. El aspirante inicia App.	1. Introducir N° Pre matricula.
2. La App muestra resultado de admisión.	2. App muestra el resultado.
3. El actor elige salir.	3. Botón salir.
Pre-condiciones: El Aspirante realizo pre matrícula correctamente.	
Post-condiciones: El estudiante consulto con éxito su resultado de admisión.	
Suposiciones: El estudiante tiene un Smartphone con conexión web y N° pre matrícula.	



5.3. Diagrama de Secuencia Estudiante UNA

5



Aspirante Nuevo ingreso

¶



5.4. Diagrama de Clase

Consultas de Calificaciones

冊



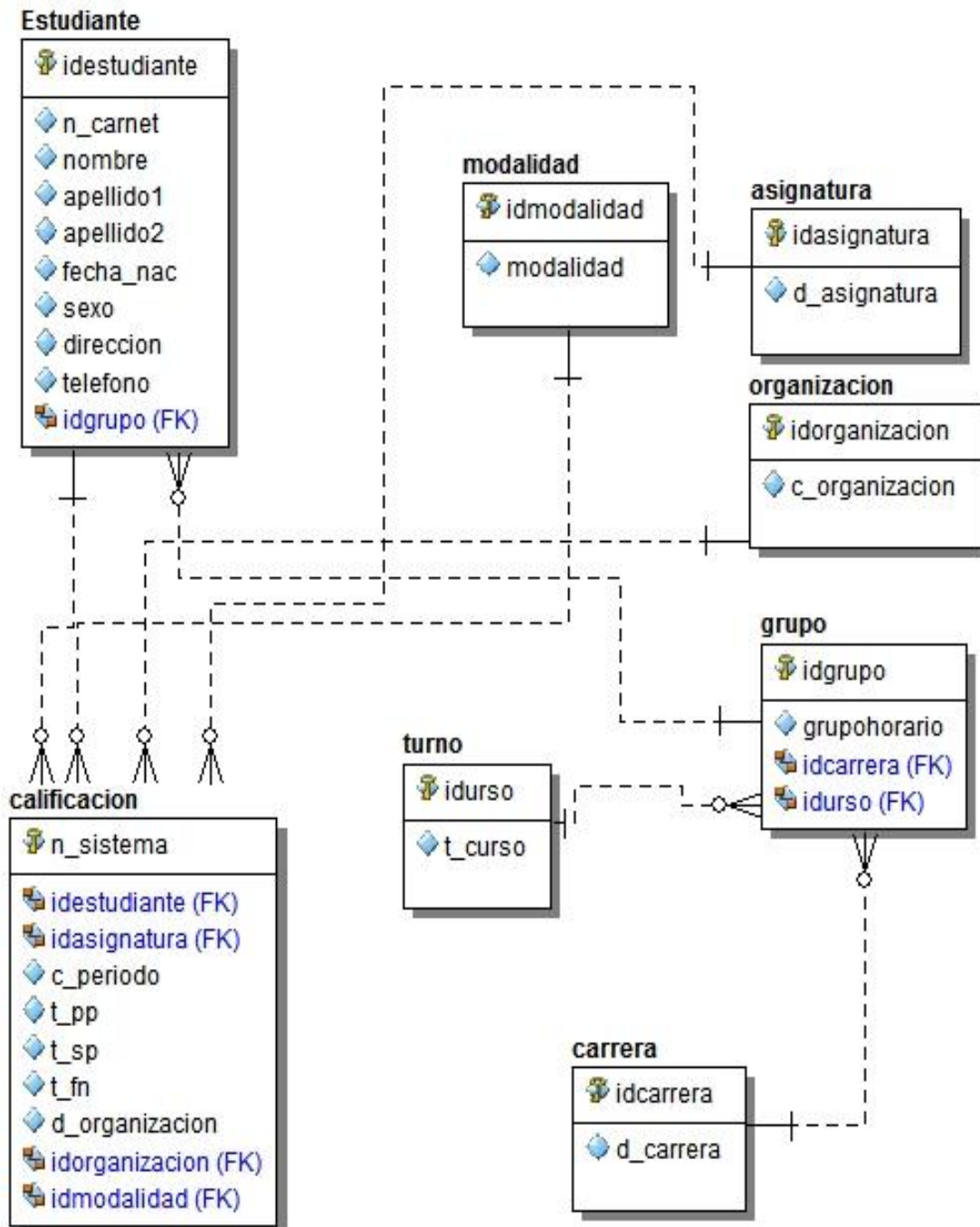
Resultado Admisión

PC5



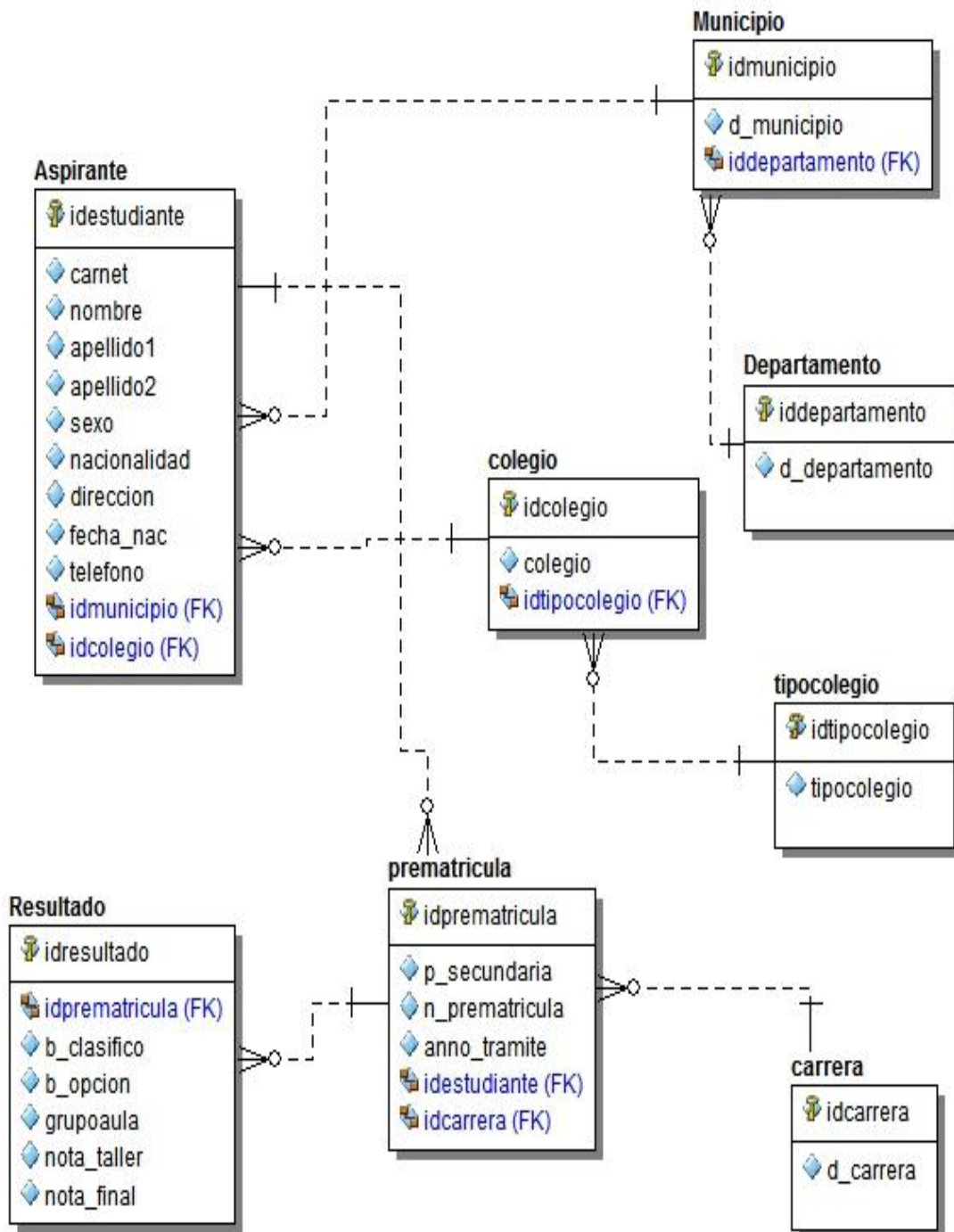
5.5. Diagrama Entidad-Relación

Consulta Calificaciones





Resultado Admisión





6. Diccionario de datos

Consultas de Calificaciones

Tabla 1 Estudiante		
Atributo	Definición	Tipo de Datos
idestudiante	Código del registro estudiante	INTEGER
n_carnet	identifica de manera única al estudiante	VARCHAR(10)
nombre	nombre del estudiante	VARCHAR(25)
apellido1	primer apellido del estudiante	VARCHAR(15)
apellido2	segundo apellido del estudiante	VARCHAR(15)
fecha_nac	fecha de nacimiento del estudiante	CHAR(10)
sexo	sexo del estudiante	VARCHAR(1)
dirección	dirección de residencia del estudiante	VARCHAR(50)
teléfono	número telefónico para contactar el estudiante	VARCHAR(8)
idgrupo	es el código de identificación de un grupo de clase de una carrera determinada	INTEGER

Tabla 2 Modalidad		
Atributos	Definición	Tipo de datos
idmodalidad	es el código de identificación de una modalidad	INTEGER
modalidad	nombre de la modalidad de una asignatura	VARCHAR(60)

Tabla 3 asignatura		
Atributo	Definición	Tipo de dato
idasignatura	es el código de identificación de una asignatura	INTEGER
d_asignatura	nombre de la asignatura	VARCHAR(150)



Tabla 4 carrera		
Atributo	Definición	Tipo de Datos
idcarrera	es el código de identificación de una carrera	INTEGER
d_carrera	nombre de la carrera	VARCHAR(100)
idurso	es el código de identificación de un turno	VARCHAR(10)

Tabla 5 turno		
Atributo	Definición	Tipo de datos
idurso	es el código de identificación de un turno	VARCHAR(10)
t_curso	nombre del turno de la carrera	VARCHAR(50)

Tabla 6 organización		
Atributo	Definición	Tipo de Datos
idorganizacion	Es el código de identificación de una facultad	VARCHAR(10)
c_organizacion	nombre de la facultad	VARCHAR(100)

Tabla 7 grupo		
Atributo	Definición	Tipo de Datos
idgrupo	es el código de identificación de un grupo de clase de una carrera determinada	INTEGER
grupohorario	nombre del grupo	VARCHAR(4)



Tabla 8
calificación

Atributo	Definición	Tipo de datos
n_sistema	es el registro del sistema	VARCHAR(9)
idestudiante	almacena el código que identifica la identidad del registro estudiante	INTEGER
idasignatura	es el código de identificación de una asignatura	INTEGER
c_periodo	es el año académico en curso	VARCHAR(4)
t_pp	primera nota parcial de una asignatura	DECIMAL(18,0)
t_sp	segunda nota parcial de una asignatura	DECIMAL(18,0)
t_fn	nota final de una asignatura	DECIMAL(18,0)
d_organizacion	nombre de los semestres	VARCHAR(3)
idorganizacion	Es el código de identificación de una facultad	VARCHAR(10)
idmodalidad	es el código de identificación de una modalidad	INTEGER

Resultados Admisión

Tabla 1
Aspirante

Atributo	Definición	Tipo de Datos
idestudiante	código de registro estudiantes	INTEGER
carnet	identificación del aspirante si llegase a clasificar	VARCHAR(10)
nombre	nombre del estudiante de primer ingreso	VARCHAR(20)
apellido1	primer apellido del aspirante	VARCHAR(20)
apellido2	segundo apellido del aspirante	VARCHAR(20)
sexo	corresponde al sexo del aspirante	CHAR(1)
nacionalidad	nacionalidad donde proviene el aspirante	VARCHAR(15)
dirección	dirección del aspirante	VARCHAR(100)
fecha_nac	fecha de nacimiento	VARCHAR(10)
teléfono	teléfono del aspirante	INTEGER
idmunicipio	código del registro municipio	INTEGER
idcolegio	código del registro colegio	INTEGER



Tabla 2 colegio		
Atributo	Definición	Tipo de Datos
idcolegio	código del registro colegio	INTEGER
colegio	nombre del colegio	VARCHAR(35)
idtipocolegio	código del registro de tipo de colegio	INTEGER

Tabla 3 tipocolegio		
Atributo	Definición	Tipo de Datos
idtipocolegio	código del registro de tipo de colegio	INTEGER
tipocolegio	tipo de colegio (estatal o privado)	VARCHAR(15)

Tabla 4 Municipio		
Atributos	Definición	Tipo de Datos
idmunicipio	código del registro municipio	INTEGER
d_municipio	nombre de un municipio	VARCHAR(30)
iddepartamento	código del registro departamento	VARCHAR(30)

Tabla 5 Departamento		
Atributo	Definición	Tipo de Datos
iddepartamento	código del registro departamento	VARCHAR(30)
d_departamento	nombre de un departamento	VARCHAR(30)

Tabla 6 carrera		
Atributo	Definición	Tipo de Datos
idcarrera	código del registro carrera	INTEGER
d_carrera	nombre de la carrera	CHAR(20)



Tabla 7
prematricula

Atributo	Definición	Tipo de Datos
idprematricula	código del registro pre matricula	INTEGER
p_secundaria	promedio de notas obtenido en secundaria	DECIMAL(18,0)
n_prematricula	código pre matricula del aspirante	VARCHAR(4)
anno_tramite	año de pre matricula realizada	INTEGER
idestudiante	código de registro estudiante	INTEGER
idcarrera	código del registro carrera	INTEGER

Tabla 8
Resultado

Atributo	Definición	Tipo de Datos
idresultado	código del registro resultado	INTEGER
idprematricula	código del registro pre matricula	INTEGER
b_clasifico	estado del resultado(si, no)	VARCHAR(2)
b_opcion	opción clasificada del aspirante	INTEGER
grupoaula	código del grupo donde realizo el propedéutico	VARCHAR(15)
nota_taller	promedio obtenido en el curso propedéutico	DECIMAL(18,0)
nota_final	nota final de resultado	DECIMAL(18,0)



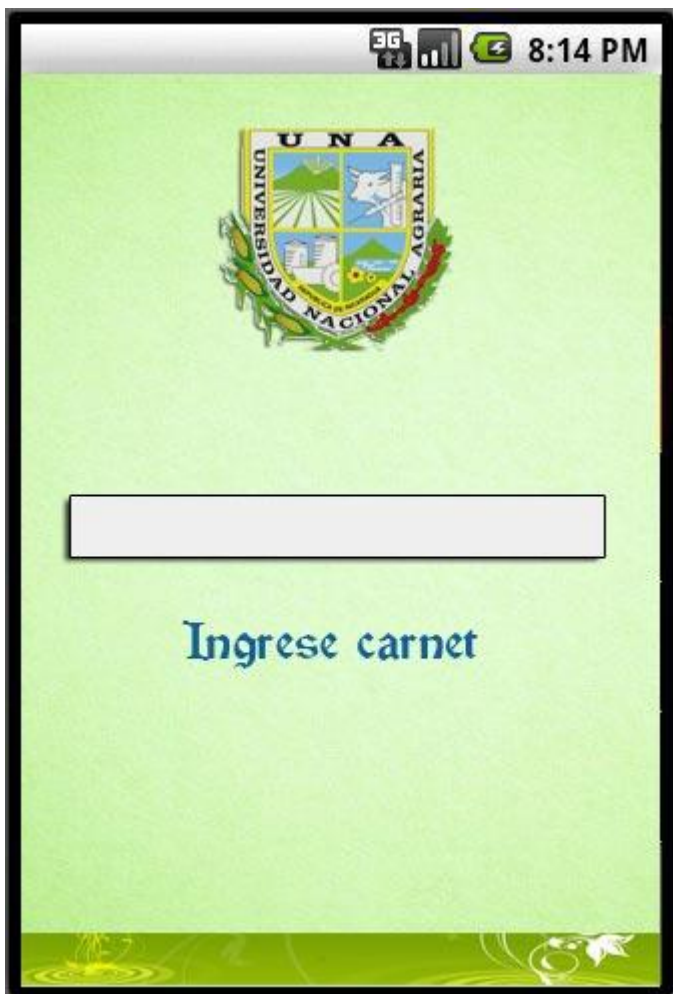
7. Pantallas de la Aplicación.

Al iniciar la aplicación en el móvil se mostrara la siguiente pantalla de inicio.



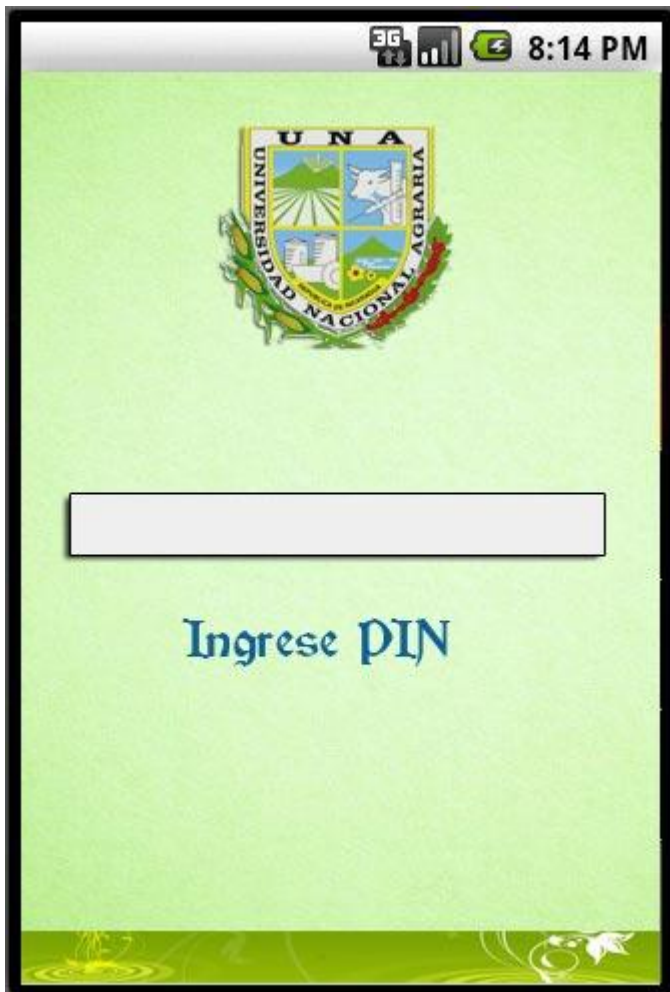


El módulo de consultas de notas, el estudiante ingresa su número de carnet





Seguidamente tiene que ingresar su número Pin para general su respectiva consulta.





Módulo de Resultado de Admisión

El aspirante sólo tiene que ingresar su número de Pre-Matricula para conocer su resultado de ingreso.





7.1. Pantallas con Datos

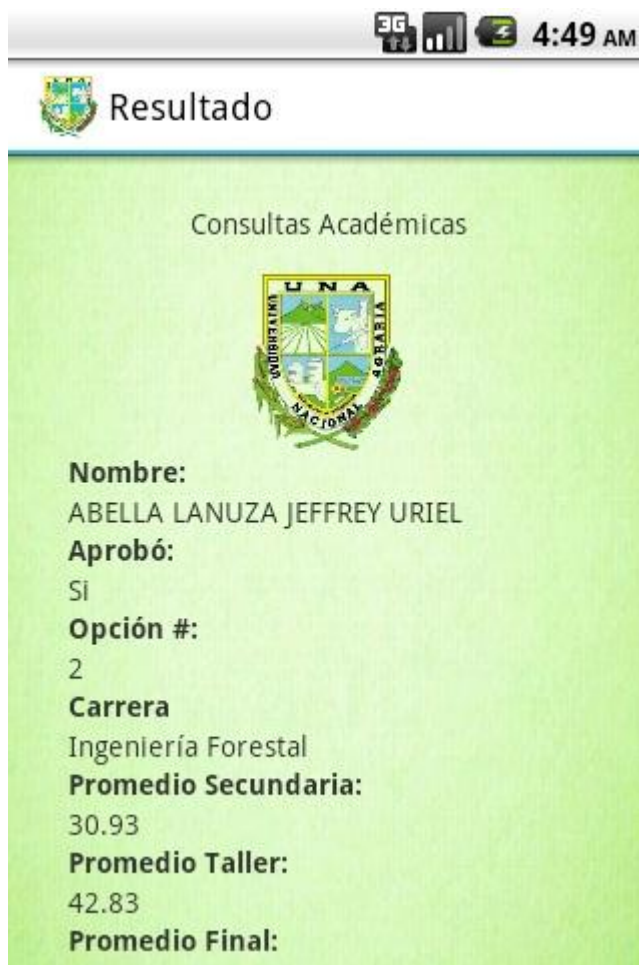
En la siguiente pantalla de consulta de calificaciones se usaron como datos

Carnet: 04010231, PIN: HGEA661900





Para la pantalla de Consulta de Resultado de Admisión se usó el siguiente Código de **Pre-Matricula: 0714**



8. Pruebas de la Aplicación.

Las pruebas se realizaron en teléfonos inteligentes con distintas versiones de Android: Motorola XT316 versión 2.3.4, Motorola Defy versión 2.3.7, LG L5 versión 4.1.2, HTC ONE versión 4.4.3 en donde accedimos a la aplicación desde distintos puntos de la capital y en distintas áreas de la universidad(UNA); en el transcurso de pruebas la aplicación no presenta ninguna falla de compatibilidad en las diferentes versiones de los móviles, se accedió a la información en tiempo óptimo de respuesta.

Para tener acceso a la aplicación hicimos uso de conexión wifi y servicios de paquetes de datos por las compañías telefónicas del país (claro y movistar).

Imágenes de los móviles de pruebas:



Motorola xt316



Motorola Defy



LG L5



HTC ONE M7

También se probó la aplicación en los móviles de los estudiantes, la aplicación se ejecutó en distintas versiones que poseían sus sistemas operativos Android, se realizaron consultas sin ningún problema de compatibilidad de versión y conexión de datos al servicio que brinda la aplicación.



VII. Conclusiones

- Se determinó las condiciones tecnológicas que posee la Universidad Nacional Agraria, en el cual la institución cuenta con la infraestructura necesaria para implementar la aplicación móvil.
- Se analizó los diagramas principales relacionados al trámite de nota y consulta de resultados de admisión de la Universidad.
- Se elaboró las pantallas de diseño de la aplicación de consulta de notas y resultado de admisión.
- Se programó una aplicación en donde los estudiantes de la Universidad Nacional Agraria, pueden consultar las notas y buscar su resultado de admisión de una manera más fácil y segura.



VIII. Recomendaciones

- ❖ Agregar otras funcionalidades relevantes a la aplicación, para que los estudiantes tengan un mejor acceso a la información académica.
- ❖ Implementar la aplicación en otras plataformas de desarrollo para cubrir la población que usa otro sistema operativo móvil.
- ❖ Garantizar el acceso a la aplicación en la comunidad estudiantil.



IX. Bibliografía

- Alarcon, R. (2000). *Diseño Orientado a Objetos con UML*. España: Grupo Eidos.
- Fouler, M. (1999). *UML gota a gota*. Mexico: addison wesley Iberoamericana.
- (2003). En I. Gilfillan, *La biblia de MySQL*. Madrid: Anaya Multimedia.
- Gonzalez, C. (1996). *Sistema de Base de Datos*. san jose: editorial tecnologica de costarica.
- Kenneth E. Kendall, J. E. (2011). *Analisis y Diseño de Sistemas*. Mexico: Pearson Educacion.
- Larman, c. (2003). *UML Y Patrones introduccion al Analisis y Diseño de Objetos*. Mexico: Prentice Hall.
- Pressman, R. (2010). *Ingenieria del software: un enfoque practico*. España: McGraw Hill.
- Silberschatz, A. (2006). *Fundamentos de Dase de Datos*. España: Mc Graw Hill.
- Sommerville, I. (2011). *Ingenieria de Software*. madrid: pearson educacion S.A.
- mredison*. (16 de Septiembre de 2014). Obtenido de <http://mredison.wordpress.com/2009/03/25/servicios-web-en-aspnet/>
- Android curso*. (s.f.). Recuperado el 16 de Septiembre de 2014, de <http://www.androidcurso.com/index.php/tutoriales-android/43-unidad-10-internet-sockets-http-y-servicios-web/334-servicios-web-basados-en-soap>
- dtecn*. (s.f.). *dtecn*. Recuperado el 2014, de <http://dtecn.com/versiones-sistema-operativo-android/>
- ecured*. (s.f.). Obtenido de www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado
- gabharan*. (14 de Diciembre de 2012). *Android*. Obtenido de <http://histinf.blogs.upv.es/2012/12/14/android/>
- ibm*. (s.f.). *ibm*. Obtenido de www.ibm.com/developerworks/ssa/library/os-ecov/



jcecebmob. (s.f.). *Sistemas Operativos Móbiles*. Recuperado el 2014, de <http://jcecebmob.blogspot.com/2012/03/mde-lab-comparativa-sistemas-operativos.html>

mastermagazine. (s.f.). *mastermagazine*. Recuperado el 2014, de <http://www.mastermagazine.info/termino/3874.php#ixzz32VPAYZtT>

ocu. (s.f.). *ocu*. Recuperado el 2014, de www.ocu.org

programacion. (s.f.). Recuperado el 16 de Septiembre de 2014, de <http://www.programacion.com.py/moviles/android/acceso-a-web-service-soap-en-android>

UNA. (s.f.). Obtenido de www.una.edu.ni

uoc. (s.f.). *www.uoc.edu*. Recuperado el 14 de junio de 2014, de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf

wikipedia. (s.f.). Recuperado el 16 de septiembre de 2014, de http://es.wikipedia.org/wiki/Simple_Object_Access_Protocol



X. Anexos

Encuesta realizada para la obtención de la información.

Universidad Nacional Autónoma de Nicaragua
Facultad de Ciencias e Ingeniería
Departamento de Computación
Seminario de Graduación



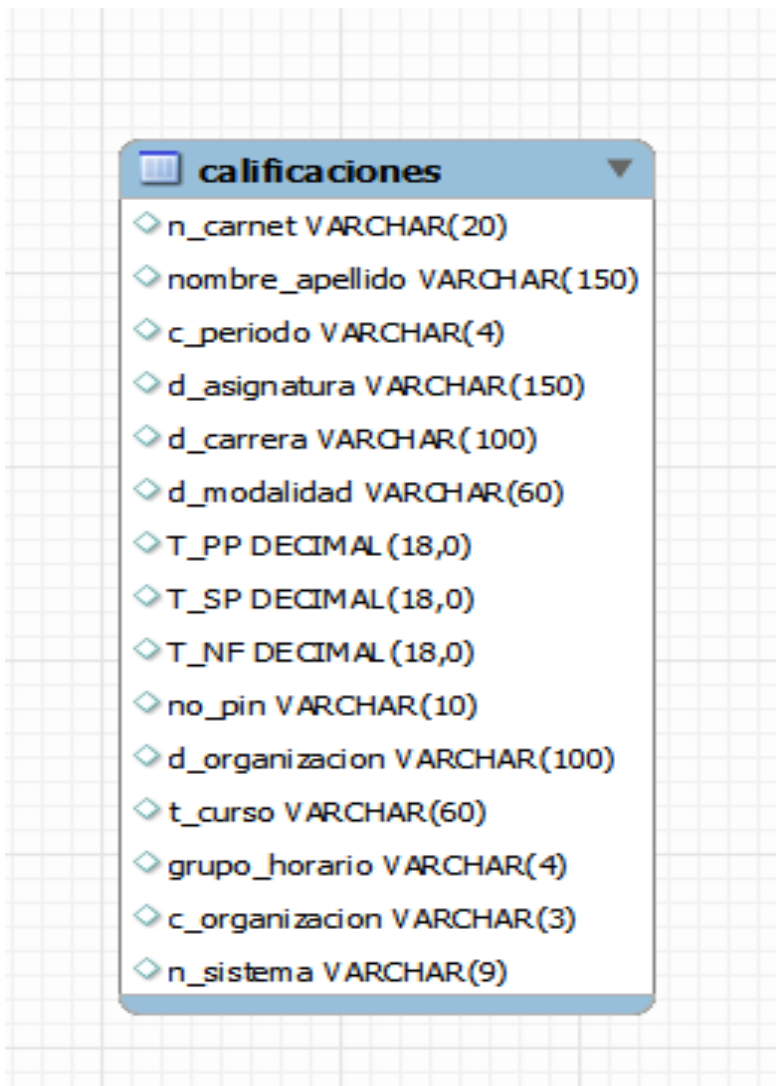
Encuesta investigativa de uso de Smartphone en Universidad Nacional Agraria

1. ¿Qué sistema operativo móvil es de su preferencia?
Android__ IOS__ Windows Phone__ BlackBerry__
2. ¿Qué sistema operativo posee tu Smartphone actualmente?
Android__ IOS__ Windows Phone__ BlackBerry__
3. ¿si se desarrollara una aplicación para Smartphone de consulta de información de la UNA, en que sistema operativo móvil preferirías que se desarrolle?
Android__ IOS__ Windows Phone__ BlackBerry__
4. ¿Qué información de consulta te resultaría más relevante en una aplicación móvil?
5. ¿Actualmente esa información de relevancia para ti es de fácil acceso?
SI__ NO__
6. ¿crees que sería menos complicado consultar esa información desde tu móvil?
SI__ NO__



Bases de Datos que nos facilitó la universidad para realización de pruebas.

BD de Registro de calificaciones académicas de los estudiantes de la Universidad Nacional Agraria.





BD de pre-Matricula de los aspirantes de nuevo ingreso en Universidad Nacional Agraria.

resultados_proped	
nombre_apellido	VARCHAR(150)
b_clasifico	VARCHAR(2)
b_opcion_clasifico	TINYINT(4)
ano_tramite	VARCHAR(4)
d_carrera	VARCHAR(100)
n_prematricula	VARCHAR(4)
n_carnet	VARCHAR(10)
grupo_aula	VARCHAR(15)
d_nacionalidad	VARCHAR(100)
b_tipo_colegio	VARCHAR(30)
p_secundaria	DECIMAL(18,2)
nota_taller	DECIMAL(18,2)
nota_final	DECIMAL(18,2)