

**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

**Centro Universitario UAEM Atlacomulco**



**“GRAFICACIÓN DE DATOS DE EXPERIMENTOS DE PLASMA EN SISTEMAS OPERATIVOS ANDROID”**

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE**

**INGENIERÍA EN COMPUTACIÓN**

**P R E S E N T A:**

RAFAEL SAMANO PICHARDO

No. De Cu. 0720028

**ASESOR:** DR. CARLOS EDUARDO TORRES REYES

**ATLACOMULCO, MÉXICO, NOVIEMBRE 2014**

ÍNDICE

[INTRODUCCIÓN 8](#_Toc404948328)

[CAPITULO I MARCO PROBLEMÁTICO 10](#_Toc404948329)

[PLANTEAMENTO DEL PROBLEMA 11](#_Toc404948330)

[PREGUNTA DE INVESTIGACIÓN 13](#_Toc404948331)

[OBJETIVO GENERAL 14](#_Toc404948332)

[OBJETIVOS ESPECÍFICOS 15](#_Toc404948333)

[HIPÓTESIS 16](#_Toc404948334)

[JUSTIFICACION 17](#_Toc404948335)

[METODOLOGÍA 18](#_Toc404948336)

[APORTACIONES 19](#_Toc404948337)

[CAPITULO II.- MARCO REFERENCIAL 20](#_Toc404948338)

[ANTECEDENTES 21](#_Toc404948339)

[PLASMA Y TIPOS DE PLASMAS 25](#_Toc404948340)

[SENSORES Y TIPOS 27](#_Toc404948341)

[GRÁFICAS Y TIPOS 28](#_Toc404948342)

[ANDROID 29](#_Toc404948343)

[ARQUITECTURA DE ANDROID 31](#_Toc404948344)

[CARACTERÍSTICAS DE SDK DE ANDROID 33](#_Toc404948345)

[SOFTWARE E INGENIERÍA DEL SOFTWARE 33](#_Toc404948346)

[INGENIERÍA DEL SOFTWARE 34](#_Toc404948347)

[METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE 34](#_Toc404948348)

[CLASIFICACIÓN DE LAS METODOLOGÍAS 35](#_Toc404948349)

[CICLO DE VIDA DEL SOFTWARE 35](#_Toc404948350)

[MODELOS DE PROCESO 37](#_Toc404948351)

[PROCESO PERSONAL DE SOFTWARE PSP 38](#_Toc404948352)

[FASES DEL PSP 39](#_Toc404948353)

[LENGUAJE DE MODELADO UNIFICADO 40](#_Toc404948354)

[SISTEMA DE INFORMACIÓN 40](#_Toc404948355)

[BASE DE DATOS 41](#_Toc404948356)

[SISTEMA ADMINISTRADO DE BASES DE DATOS 42](#_Toc404948357)

[XLM 43](#_Toc404948358)

[JAVA 44](#_Toc404948359)

[CAPITULO III ANÁLISIS DE LA APLICACIÓN 45](#_Toc404948360)

[ANÁLISIS DEL SISTEMA 46](#_Toc404948361)

[INGENIERÍA DE REQUERIMIENTOS 46](#_Toc404948362)

[IEEE 830 PRACTICA RECOMENDADA PARA REQUISITOS DE ESPECIFICACIONES DE SOFTWARE 46](#_Toc404948363)

[PROPÓSITO DEL SRS 47](#_Toc404948364)

[ALCANCE DEL SRS 47](#_Toc404948365)

[PERSONAL INVOLUCRADO 48](#_Toc404948366)

[PERSPECTIVA DEL PRODUCTO 49](#_Toc404948367)

[FUNCIONALIDAD DEL PRODUCTO 49](#_Toc404948368)

[CARACTERÍSTICAS DEL USUARIO 50](#_Toc404948369)

[RESTRICCIONES 51](#_Toc404948370)

[SUPOSICIONES Y DEPENDENCIAS 52](#_Toc404948371)

[EVOLUCIÓN PREVISIBLE DEL SISTEMA 52](#_Toc404948372)

[REQUISITOS ESPECÍFICOS 53](#_Toc404948373)

[REQUISITOS COMUNES DE LAS INTERFACES 55](#_Toc404948374)

[INTERFACES DE USUARIO 55](#_Toc404948375)

[INTERFACE DE HARDWARE 56](#_Toc404948376)

[INTERFACE DE COMUNICACIÓN 56](#_Toc404948377)

[REQUISITOS FUNCIONALES 57](#_Toc404948378)

[REQUISITO FUNCIONAL NÚMERO UNO 57](#_Toc404948379)

[REQUISITO FUNCIONAL NÚMERO DOS 58](#_Toc404948380)

[REQUISITO FUNCIONAL NÚMERO TRES 60](#_Toc404948381)

[REQUISITO FUNCIONAL NÚMERO CUATRO 61](#_Toc404948382)

[REQUISITO FUNCIONAL NÚMERO CINCO 62](#_Toc404948383)

[REQUISITO FUNCIONAL NÚMERO SEIS 63](#_Toc404948384)

[REQUISITO FUNCIONAL NÚMERO SIETE 64](#_Toc404948385)

[REQUISITOS NO FUNCIONALES 65](#_Toc404948386)

[REQUISITOS DE RENDIMIENTO 65](#_Toc404948387)

[SEGURIDAD Y VALIDACIÓN DE INFORMACIÓN 66](#_Toc404948388)

[DISPONIBILIDAD E INSTALACIÓN 66](#_Toc404948389)

[CAPITULO IV MEDICIÓN Y MODELADO DE LA APP 67](#_Toc404948390)

[DISEÑO DEL SISTEMA 68](#_Toc404948391)

[CASOS DE USO 68](#_Toc404948392)

[DIAGRAMA DE SECUENCIA 70](#_Toc404948393)

[DIAGRAMA DE ACTIVIDAD 74](#_Toc404948394)

[DIAGRAMA DE CLASES 78](#_Toc404948395)

[MODELO CONSTRUCTIVO DE COSTOS II 79](#_Toc404948396)

[ESTIMACIÓN DE LA APLICACIÓN 79](#_Toc404948397)

[CAPITULO V DISEÑO DE LOS COMPONENTES DE LA APLICACIÓN 93](#_Toc404948398)

[DISEÑO DE BASE DE DATOS 94](#_Toc404948399)

[MODELO DE ENTIDAD RELACIÓN 94](#_Toc404948400)

[DICCIONARIO DE DATOS 95](#_Toc404948401)

[DISEÑO VISUAL Y SECUENCIA DE GESTOS 97](#_Toc404948402)

[WIREFRAME DE INICIO 97](#_Toc404948403)

[WIREFRAME DE ALTA 98](#_Toc404948404)

[WIREFRAME CENTRAL 99](#_Toc404948405)

[WIREFRAME DE RESULTADOS SQL 100](#_Toc404948406)

[WIREFRAME DE GRAFICACIÓN 101](#_Toc404948407)

[CAPÍTULO VI PRUEBAS DE LA APLICACIÓN 103](#_Toc404948408)

[PRUEBAS DE SOFTWARE 104](#_Toc404948409)

[PLANTILLAS DE PRUEBAS DE SOFTWARE 104](#_Toc404948410)

[PLANTILLA DE AGREGAR DATOS DEL SMDB 104](#_Toc404948411)

[PLANTILLA DE EDITAR Y ELIMINAR DATOS 110](#_Toc404948412)

[PLANTILLA DE ACTIVIDAD CENTRAL 115](#_Toc404948413)

[PLATILLA DE TABLA SQL 119](#_Toc404948414)

[PLANTILLA DE GRAFICA 124](#_Toc404948415)

[COMPARATIVA DE SQLGRAPHIC CON EXCEL 128](#_Toc404948416)

[CONCLUSIONES 130](#_Toc404948417)

[REFERENCIA BIBLIOGRÁFIC 131](#_Toc404948418)

[ANEXOS 140](#_Toc404948419)

[ANEXO 1: DOMINIOS DE APLICACIÓN DE SOFTWARE 141](#_Toc404948420)

[ANEXO 2: ACTIVIDADES DE LA INGENIERÍA DEL SOFTWARE 143](#_Toc404948421)

[ANEXO 3: ACTIVIDADES ESTRUCTURALES DEL PROCESO DE INGENIERÍA DE SOFTWARE 144](#_Toc404948422)

[ANEXO 4 PROCESOS DEL CICLO DE VIDA DEL SOFTWARE 145](#_Toc404948423)

[ANEXO 5 TAREAS DE LA INGENIERÍA DE REQUERIMIENTOS 147](#_Toc404948424)

[ANEXO 6 IEEE 830 REQUISITOS DE ESPECIFICACIONES DE SOFTWARE 149](#_Toc404948425)

[ANEXO 7 EL MODELO CONSTRUCTIVO DE COSTOS O COCOMO II 153](#_Toc404948426)

**ÍNDICE DE FIGURAS**

**Capítulo II**

[Figura 2. 1: Clasificación de las graficas 28](file:///D:\UAEM\TESIS\Rafael%20Samno%20Pichardo%20GRAFICACIÓN%20Y%20REPRESENTACIÓN%20EN%20ANDROID%20-%20TESIS%20v3.docx#_Toc404000032)

[Figura 2. 2: Arquitectura del sistema operativo Android 32](#_Toc404000033)

[Figura 2. 3: Diagrama de procesos del ciclo de vida del software 36](#_Toc404000034)

[Figura 2. 4: Diagrama de las etapas del proceso personal de software. 39](#_Toc404000035)

[Figura 2. 5: Componentes de un sistema de información. 41](file:///D:\UAEM\TESIS\Rafael%20Samno%20Pichardo%20GRAFICACIÓN%20Y%20REPRESENTACIÓN%20EN%20ANDROID%20-%20TESIS%20v3.docx#_Toc404000036)

[Figura 2. 5: Componentes de un sistema de información. 41](file:///D:\UAEM\TESIS\Rafael%20Samno%20Pichardo%20GRAFICACIÓN%20Y%20REPRESENTACIÓN%20EN%20ANDROID%20-%20TESIS%20v3.docx#_Toc404000037)

**Capítulo IV**

[Figura 4. 1 - Diagrama de casos de uso de la Aplicación SQLGraphic. 69](#_Toc404945082)

[Figura 4. 2 - Subdiagrama de casos de uso de Aplicación SQLGraphic. 70](#_Toc404945083)

[Figura 4. 3 - Diagrama de secuencia de autenticación en el servidor. 71](#_Toc404945084)

[Figura 4. 4 - Diagrama de elección de la base de datos. 72](#_Toc404945085)

[Figura 4. 5 - Diagrama de secuencia de sentencia SQL. 73](#_Toc404945086)

[Figura 4. 6 - Diagrama de sentencia de graficación. 74](#_Toc404945087)

[Figura 4. 7 - Diagrama de actividad para agregar datos del servidor a la BD local. 75](#_Toc404945088)

[Figura 4. 8 - Diagrama de actividad de autenticación al servidor. 75](#_Toc404945089)

[Figura 4. 9 - Diagrama de actividad de selección de BD. 76](#_Toc404945090)

[Figura 4. 10 - Diagrama de actividad de sentencia de tabla. 77](#_Toc404945091)

[Figura 4. 11 -Diagrama de actividad de sentencia de graficación. 77](#_Toc404945092)

[Figura 4. 12 - Diagrama de clases. 78](#_Toc404945093)

**Capítulo V**

[Figura 5. 1 - Diagrama De Entidad Relación Notación Tipo Chen. 94](#_Toc404945075)

[Figura 5. 2 - Diagrama De Entidad Relación Notación Tipo “Pata de Gallo”. 95](#_Toc404945076)

[Figura 5. 3 - Wireframe de inicio de SQLGraphic. 98](#_Toc404945077)

[Figura 5. 4 - Wireframe de alta de la aplicación SQLGraphic. 99](#_Toc404945078)

[Figura 5. 5 - Wireframe central de la aplicación SQLGraphic. 100](#_Toc404945079)

[Figura 5. 6 - Wireframe de resultados SQL de la aplicación SQLGraphic. 101](#_Toc404945080)

[Figura 5. 7 - Wireframe de graficación de la aplicación SQLGraphic. 102](#_Toc404945081)

**Capítulo VI**

[Figura 6. 1 – Resultado de la tabla de prueba 7 109](#_Toc404940018)

[Figura 6. 2 - Resultado de la tabla de prueba 11. 114](#_Toc404940019)

[Figura 6. 3 – Resultado de la prueba 17 122](#_Toc404940020)

[Figura 6. 4 – Resultado de la tabla de prueba 22 127](#_Toc404940021)

[Figura 6. 5 - Serie de plasma graficada en Excel. 128](#_Toc404940022)

# Introducción

El ser humano se mueve rápido y vivimos en un mundo en que la pérdida de un segundo puede ser vital en muchos sentidos, un mundo en el que el hombre pretende llegar a todas partes sin ser un dios... pero, ante todo, se busca satisfacer las necesidades de un individuo dentro de una sociedad, lo que ha llevado a integrar al hombre y a la maquina en un mismo ente, como dos partes de un todo.

La presente investigación se refiere al tema de graficación de datos en sistemas operativos Android, en donde el objetivo principal es desarrollar una aplicación que sea compatible con el sistema operativo (SO) Android, para la representación de datos adquiridos de una simulación de plasma no térmico.

En ese orden de ideas, se define una gráfica como una representación de datos, generalmente numéricos, para mostrar de manera visual la relación entre dos variables. La construcción de un gráfico es uno de las formas más directas de localizar o verificar la relación entre dos conjuntos de valores experimentales. Es frecuente observar que el valor de una variable de un sistema se modifica simultáneamente con el cambio del valor adoptado por otra variable como es: la velocidad con respecto al tiempo, la probabilidad de lluvia en relación a la presión atmosférica, etc.

A mayor abundamiento, Android, es la primera plataforma abierta y comprensible para dispositivos móviles, esto incluye un sistema operativo, una interfaz de usuario y aplicaciones, todo el software necesario para el funcionamiento de un teléfono móvil, pero sin obstáculos de propiedad que dificultan la innovación móvil.

Los datos a representar en esta investigación son obtenidos a través de sensores de una simulación de plasma no térmico y almacenados en diferentes tablas que se encuentran alojadas en una base de datos. La aplicación móvil permitirá la conexión a la base de datos y de esta manera el usuario podrá seleccionar los campos de las tablas a graficar.

La distribución de los diversos temas que estructuran el trabajo de investigación es al tenor siguiente:

Capítulo I en el cual se realizara un resumen del trabajo de investigación, una introducción, el estado del arte, el planteamiento del problema, los objetivos que persigue la investigación, la pregunta de investigación y la hipótesis de la investigación.

Capítulo II abarca términos sobre plasma, sensores, graficación, ingeniería de software, base de datos, gestores de base de datos, procesos de desarrollo de software y descripción sistema operativo Android.

Capítulo III se enfoca en el análisis y diseño de la aplicación móvil, se utilizara una plantilla SRS para la obtención de requerimientos funcionales y no funcionales, perspectivas, limitaciones, alcance, personas involucradas en el desarrollo del sistema, entre otras características.

Capítulo IV se utiliza COCOMO II para realizar una predicción del tamaño, esfuerzo, número de personas requeridas, tiempo para el desarrollo del software, conjuntamente se modela sistema a través de diagramas UML.

Capítulo V enfocado en el diseño de la base de datos manejando diagramas de entidad relación y de objetos. Se diseñan también los diferentes wireframe que integraran la aplicación.

Capítulo VI se readaptan y ejecutan los casos de pruebas diseñados para el sistema SQLGraphic, con el fin de corroborar que el sistema cumpla con los requisitos funcionales y no funcionales.

# Capítulo I - Marco Problemático

## Planteamiento Del Problema

La investigaciones y experimentación forman parte del nivel de progreso de la humanidad, dentro de la industria existen múltiples investigaciones y procesos que generan inmensas cantidades de datos, una de las áreas de investigación que está sobresaliendo cada vez más es la referente a la física del plasma. En México una de las instituciones que recientemente se ha enfoca en el estudio de física de plasma es la UAEM (Universidad Autónoma del Estado de México), con investigaciones encaminadas a procesos de plasmas no térmicos.

Para el estudio de este estado de la materia se emplean diferentes sensores, los cuales son dispositivos que producen una señal en respuesta a su detección o medida de una propiedad como posición, fuerza, toque, presión, temperatura, humedad, velocidad, aceleración o vibración. La tecnología de los sensores es esencial para la correcta adquisición de los datos, estos datos son resguardados en tablas que conforman una base de datos (Pallás, 2003).permitiendo

Para el correcto análisis de la investigación se emplean gráficas para una interpretación rápida de la información, además ofrece la capacidad de analizar el comportamiento de eventos, estadísticas, procesos, operaciones matemáticas, entre otras, además de tener relación con diferentes contenidos matemáticos como álgebra, trigonometría, geometría, cálculo, probabilidad y estadística (Gil & Rodríguez, 2001). Actualmente existen distintos software de aplicación que permite el estudio de procesos como los referentes a plasma no térmicos por medio del manejo y graficación de cómo son MATLAB, SIMULINK, LABVIEW, por mencionar algunos, pero todas se encuentran limitadas a la robustez del equipo en que se hayan instaladas, sea una computadora de escritorio o en su defecto una laptop, los cables o cargadores presentan otra desventaja puesto que el tiempo de conexión o de trabajo depende de la batería. La movilidad es otro de los puntos en contra que se le puede atribuir a dichos equipos, ya que aunque se consideren equipos portátiles, a su vez no lo son, siempre se encontrara con cierta incomodidad, si se intenta utilizar en algún lugar diferente a un escritorio; por lo concerniente no se puede tener acceso a los datos sin estar frente a la PC; es por ello que los dispositivos móviles presentan una opción para el investigador por sus características que presentan como son procesadores desde 1.2Ghz de doble núcleo e inclusive de 1.5Ghz de cuatro núcleos, capacidad de almacenamiento de hasta 32 Gb en micro SD, acceso a redes móviles, correo, noticias, páginas web, portabilidad, facilidad de uso, el tiempo de arranque más rápido, liviandad, miles de aplicaciones para los diferentes sistemas operativos móviles (IOs, Android, Windows Phone), permite realizar llamadas y enviar SMS, entre otras (M Prieto & E. Arrieta, 2012).

Por lo antes mencionado se propone el desarrollo de una aplicación nativa en sistemas operativos Android, para el manejo y graficación de datos adquiridos de un proceso de plasma no térmico, usando éste como ejemplo de aplicación, ya que esta herramienta ayudará a la interpretación de la información, también de tener disponible la información en cualquier momento, situación y lugar, ya que con el cambio constante han surgido nuevas Tecnologías de Información (TIC’s), como lo son las redes de comunicaciones inalámbricas, base de datos, telecomunicaciones, entre otras. Emplear estas características de las TIC para acceder y representar los datos de una investigación más aun cuando en la actualidad el uso de los sistemas móviles de comunicación y acceso a Internet está creciendo fuertemente, tomando una gran importancia debido a las mejoras en las estructuras tecnológicas a nivel de comunicaciones. Dado que acceder a través de un dispositivo móvil a internet antiguamente era poco posible, debido a que las comunicaciones móviles se caracterizaban por tener poco ancho de banda, hoy este marco de referencia ha cambiado con respecto a la tecnología inalámbrica móvil, por lo que los dispositivos móviles son empleados en diferentes áreas como son: medicina, industria, educación, videojuegos, entre otros, ya que evolucionaron de un dispositivo de comunicación a un dispositivo inteligente (Barrón, Zatarain y Santillán, 2011).

## Pregunta De Investigación

¿Con el diseño e implementación de una aplicación en sistemas móviles Android se permitirá la visualización de datos adquiridos de un proceso de plasma no térmico?

## Objetivo General

Codificar una aplicación nativa del sistema operativo Android, para la manipulación y representación de datos adquiridos de un proceso de plasma no térmico.

## Objetivos Específicos

1. Analizar requerimientos funcionales y no funcionales del sistema de graficación. Para determinar las necesidades del sistema y posteriormente extraer los casos de uso
2. Realizar la plantilla de SRS (*Software Requirements Specification*) en base a las normas planteadas en IEEE 830.
3. Modelar del sistema de graficación por medio del Modelado de Agentes Móviles utilizando UML (*Lenguaje de Modelado Unificado*) extendido.
4. Usar HTML, CSS, XML, JAVA o cualquier tecnología necesaria para el desarrollo de la aplicación móvil y lograr la compatibilidad de la aplicación nativa en Android.

## Hipótesis

Por medio de una aplicación basada en el sistema operativo Android, se permitirá la transmisión y graficación de datos, para proporcionar ayuda en el estudio de un proceso plasma no térmico.

## Justificación

Se propone desarrollar un sistema en un entorno móvil para el manejo y graficación de datos adquiridos de un proceso de plasma no térmico, para optimizar la eficacia y flexibilidad en el manejo de la información, ya que la expresión numérica permite un análisis objetivo que apoye o rechace la hipótesis de un trabajo de investigación. Aun cuando la aplicación será empleada en dicho proceso, será capaz de acceder a la información alojada en una base de datos y graficarla sin tener en cuenta si pertenece o no al proceso de plasma, a través de las credenciales de autenticación que el usuario maneje en la base de datos. Se busca alcanzar una transmisión de datos, ya que siendo un dispositivo móvil puede que no se logre en todos los caso un acceso total de la información, debido a distintos factores tales como ancho de banda, cantidad de datos, tamaño de la pantalla del dispositivo móvil por mencionar algunos. Además se empleara como SO Android ya que para desarrollar en dicho sistema no es necesario de la adquisición de una licencia de Developer como es el caso de Apple o de Windows Phone, solo si se deseara poner a disposición del público en Google Play, otro de los beneficios son las diversas plataforma de desarrollo que ofrece como son Eclipse, NetBeans, entre otras, ya que al contar con una licencia de tipo Open Source no tiene costo alguno, a diferencia del Visual Studio el cual es indispensable para programar en Windows Phone.

## Metodología

Las etapas del método científico aplicables a este desarrollo del presente documento son las siguientes:

**Primera fase**: mediante la investigación documental se realizara la búsqueda de material bibliográfico para la obtención de información acerca de desarrollo de aplicaciones para Android y de herramientas opcionales para el desarrollo de aplicaciones móviles.

**Segunda fase**: uso de metodología exploratoria para obtener información acerca de sistemas o aplicaciones que permitan conectarse de manera remota a una base de datos, para generar en base a la información recolectada los siguientes puntos.

* Plantilla de requisitos de especificaciones de software: teniendo como propósito definir los requerimientos del sistema, para el desarrollo de una aplicación nativa en Android que permitirá la graficación de datos alojados en DB.
* Estimación de costes: por medio de modelo constructivo de costos se analizará el tamaño en líneas de código posibles a tener, el esfuerzo y el tiempo que se invertirán en la realización de la aplicación.

**Tercera fase**: se usara la metodología analítica para realizar el diseño y análisis de la aplicación móvil, auxiliándose de UML para modelar las distintas clases del programa mediante el uso de Agentes Móviles.

**Cuarta fase**: aplicación de métodos y técnicas de la metodología del proceso personal de software. El cual es un proceso de auto mejora que ayuda a controlar, administrar y mejora la forma de trabajar las actividades de la ingeniería de sistemas, además de ser considerado un marco estructurado para el desarrollo de software.

## Aportaciones

1. Esta investigación tendrá un impacto en el estudio de la telefonía móvil y en el acceso a la información.
2. Dentro del área de conocimiento de Desarrollo de Sistemas, una implementación de métodos con base a métricas de software comprobando de esta manera su efectividad de acuerdo al sistema propuesto.
3. En las distintas áreas de investigación al generar una forma diferente para tener acceso en todo momento de los datos, por medio de gráficas y tablas.

# Capítulo II - Marco Referencial

## Antecedentes

El mundo de la tecnología se encuentra inmerso en un proceso de transformación constante, la graficación se utiliza en diversos procesos industriales, económicos, militares, de investigación científica, por nombrar algunos. La integración entre la graficación y los dispositivos móviles ya es un hecho, al grado de poder producir, distribuir y exhibir la información mediante innovadoras aplicaciones móviles, las cuales son conscientes del entorno físico en el que están siendo usadas, logran adaptarse adecuadamente a los servicios que se ofrecen, consiguiendo reducir la brecha existente entre el mundo real y el mundo virtual de los sistemas de información.

La manipulación y graficación de datos es de vital importancia para el análisis de resultados de toda investigación experimental, típicamente la manipulación de datos numéricos es almacenada en tablas, sin embargo una gráfica permite una interpretación rápida de la información, además de ofrecer la capacidad de analizar el comportamiento de un evento, estadísticas, procesos, operaciones matemáticas, entre otras (Gil & Rodríguez, 2001).

La graficación por computadora tiene sus orígenes en el año de 1950 con el surgimiento de las computadoras digitales y en el año de 1959 surgió el primer sistema de dibujo por computadora, el DAC-1(Design Augmented by Computers) que fue creado por General Motors e IBM[[1]](#footnote-1), años después fue sustituido por CADANCE (James, 1996).

Después en 1961 se desarrolló Sketchpad por Ivan Edward Sutherland, un programa de dibujo informático el cual permitía dibujar figuras simples en la pantalla por medio de una pluma de luz. En los años posteriores Computer Image Corporation (CIC), desarrolló sistemas complejos, tales como ANIMAC, SCANIMATE y CAESAR que abrieron nuevas fronteras en la graficación por computadora (Francis & Stephen, 2007).

En el año de 1978 el laboratorio central de Física Aplicada de la Universidad John-Hopkins publica un trabajo "Matematical elements for computer graphics" de David F. Rogers, siendo de importancia tanto para el trazado de líneas como en la representación de objetos naturales por computadora (James, 1986). Con el paso del tiempo la computación gráfica evoluciona a tal nivel que en actualidad la graficación se utiliza de manera rutinaria en diversas áreas como es en las ciencias exactas, ingeniería, dentro de empresas e industrias, en el sector gubernamental, en el desarrollo de videojuegos, en publicidad, educación, solo por mencionar algunas (Francis & Stephen, 2007).

Hoy por hoy existen diferentes aplicaciones que permiten graficar datos de toda índole por medio de la computadora, smartphone o tablet. Sin embargo afirma la consultora Gartner “las computadoras están siendo sustituidas por celulares inteligentes, ya que en el último trimestre del 2011 se vendieron 419 millones de unidades móviles”, aun cuando las ventas mundiales a usuarios finales descendieron un 2,3% en el segundo trimestre de 2012 (TICbeat, 2012). Los teléfonos celulares proporcionan diversos servicios electrónicos, redes sociales, comunicación por SMS[[2]](#footnote-2), comunicación verbal y a larga distancia, procesamiento de datos, entretenimiento, entre otras.

La telefonía móvil como la conocemos hoy en día tiene su origen en el año de 1880, cuando Heinrich Rudolf Hertz inicia con los primeros trabajos de radiocomunicación móvil, años después, en 1897 Marconi logrando la trasmisión por radio hacia un barco, pero no fue hasta 1921 cuando se instaló el primer sistema de radiotelefonía móvil en la ciudad de Detroit, operando en una banda de 2MHz[[3]](#footnote-3) (Lara, 2002).

Hacia fines de la segunda Guerra Mundial, se introdujeron muchos otros sistemas de comunicación móvil, trabajando en frecuencias menores a los 460 MHz y ofrecían servicio a departamentos de gobierno, a la industria, al sistema de transporte y también a usuarios privados. Esta tecnología fue aprovechada en los años 50 y 60 para crear dispositivos de radio y comunicación a distancia (Walkie-Talkies). Los estándares de la G0 (Generación 0) fueron PTT[[4]](#footnote-4) (Push To Talk) y estándar IMTS por sus siglas Improved Mobile Telefone System (Rey, 1998).

La telefonía móvil de primera generación o 1G surgen en el año de 1973, con un tamaño y peso inmanejables, los móviles de primera generación fusionaban de manera analógica, los estándares que enmarcan esta generación son el NMT[[5]](#footnote-5) (Nordic Mobile Telephone) y AMPS (Advaced Mobile Phone System) (Serope, Steven y Ulises, 2002).

A inicios de los años 90’s surge la tecnología móvil denominada 2G o segunda generación, caracterizada por el paso de la telefonía analógica a la digital, a través de una serie de protocolos se mejora el manejo de llamadas, enlaces simultáneos en un ancho de banda; destaca el protocolo de servicio de mensajes cortos o SMS (Short Message Service). En 1993 el sistema celular GSM, comienza su instalación y para finales del 1994 tenía dos millones de conexiones a la central telefónica en toda Europa, el sistema fue aceptado en varias naciones no europeas. Los estándares que destacan en esta generación son el Sistema Global para Comunicaciones Móviles o denominado GSM (Global System for Mobile Communications), Acceso Múltiple por División de Código o nombrado CDMA (Code Division Multiple Access) y Servicio General de Radio por Paquetes o designado como General Packet Radio Service GPRS (Rey, 1998; Serope, et al. 2002).

A finales del año 2000 surge la tercera generación móvil o 3G, se define la posibilidad de transferir datos en movimiento. Algunas características de esta tecnología son la fotografía digital, posibilidad de grabar videos y mandarlos con un sistema de mensajería instantánea, con el paso de los años se implementaron nuevos servicios como juegos 3D, sonido mp3 o poder mantener conversaciones por videoconferencia debido a la tasa de transferencia de datos aceptable, hoy en día se pueden alcanzar velocidades superiores a los 3 Mbit/s por usuario móvil, brindando un soporte para Internet correctamente implementado (correo electrónico, descargas, etc.), el estándar que caracteriza esta generación es UMTS (Universal Mobile Telecommunications System) o Servicios Universales de Comunicaciones Móviles (Serope, et al. 2002).

Después del surgimiento de la tecnología 3G la búsqueda por desarrollar herramientas para una completa interacción con el usuario ha llevado a la creación de sistemas operativos multitarea. Los teléfonos móviles usan numerosos sistemas operativos en la actualidad, tales como: Symbian OS, Microsoft Windows Phone OS, Mobile Linux, iPhone OS (Basado en Mac OS X), Moblin (from Intel) y muchos otros sistemas operativos de propietarios. Android sistema operativo móvil propiedad de Google a alcanzó una cuota de mercado mundial del 50,9% durante el cuarto trimestre de 2011, lo cual lo lleva a ser una herramienta al alcance de la mayoría de la gente. El desarrollo de la plataforma Android se remonta al año 2005, después de que Google adquiere la compañía de Android Inc. Dos años más tarde se muestra las primeras características de SDK (Software Development Kit) de Android y en septiembre de 2008, T-Mobile anunció la disponibilidad de T-Mobile[[6]](#footnote-6) G1, el primer teléfono inteligente basado en plataforma Android. Desde entonces ha surgido el SDK 2.0, 3.0 y 4.0 ahora más o menos uno cada año (Satya & Dave, 2011). Gracias a aparición de sistemas operativos como Android se han podido realizar diversas investigaciones, con el propósito de crear aplicaciones para dispositivos móviles como las que se describen en los siguientes artículos:

En 2010, Hernández y Gaona presentan el artículos “Prototipo WAP Aplicado a un Ambiente de Aprendizaje virtual (m-learning) con estándares para el desarrollo de Aprendizaje Móvil”, con la finalidad de presentar la importancia de la educación virtual actual, en un ambiente de enseñanza como lo es m-learning que define un aprendizaje a través de dispositivos móviles que se pueden conectar y acceder a la información en línea en cualquier momento, situación y/o lugar. M-LEARNING se define como un tipo de aprendizaje que tiene como objetivo principal el uso de la tecnología inalámbrica móvil.

Por otra parte, en el artículo “Adaptación Dinámica de Contenidos Educativos en Dispositivos Móviles” se propone un sistema dinámico/adaptativo de interfaces para la visualización de cursos de aprendizaje móvil. Empleando como herramienta de creación de cursos móviles MLTutor (Barrón, Zatarain y Santillán, 2011).

## Plasma Y Tipos De Plasmas

El plasma es considerado el cuarto estado de la materia y se justifica por el hecho de que el 99 % de la materia existente en el universo se encuentra en este estado (Maher, Pierre y Emi, 1994). Un plasma es un gas de partículas cargadas donde el balance de las mismas está casi equilibrado, por lo que se considera un medio neutro. Se constituye de iones (dotados de carga eléctrica positiva o negativa), electrones libres, átomos y moléculas (partículas eléctricamente neutras) (Francis, 1984; Gordillo, 2008).

En términos generales, los plasmas se pueden dividir en dos grupos principales, los plasmas de alta temperatura o de fusión y los de bajas temperatura. Una clasificación típica y de los diferentes tipos parámetros de plasmas se muestra en la tabla 2.1. El plasma de alta temperatura implica todas las especies de electrones, iones y especies neutras, las cuales están en un estado de equilibrio térmico. Mientras que el plasma a baja temperatura se subdivide en plasma de cuasi equilibrio termodinámico local (LTE) y el plasma no térmico (NTP) (Vijay & Ashok 2008).

Tabla 2. - Clasificación del plasma.

|  |  |  |
| --- | --- | --- |
| Tipo Plasma | Descripción | Ejemplo (s) |
| Plasma De Alta Temperatura  (Plasma de Equilibrio Térmico) | La temperatura o energía media de las partículas que lo forman es o no la misma para cada tipo de partícula | * Plasma de fusión laser. * Plasma de los interiores. * Plasmas termonucleares de deuterio, tritio e impurezas de bajo número atómico (carbono) generados en los dispositivos experimentales de fusión nuclear controlada (JET, ITER) |
| Plasma A Baja Temperatura | | |
| Plasma de baja temperatura  (Plasma de Equilibrio-Cuasi) | El electrón, el ion, y las temperaturas neutrales son similares. | * La pulverización de plasma (revestimiento). * Térmica química de plasma y deposición física de vapor. * Química analítica * Grabado por plasma térmico, la sintonización, y la soldadura |
| Plasmas No Térmicos  (Plasmas Fríos) | La temperatura de las partículas neutras y los iones es de 25 a 100 grados centígrados.  La temperatura electrónica es, mucho mayor de entre 5000 y 105 grados centígrados.  Los plasmas fríos suelen producirse a baja presión (p < 133 mbar). | * Plasmas mediante sistemas de corriente continua, radiofrecuencia, microondas o en descargas pulsadas. * Plasmas fríos de corona y de barrera dieléctrica. |

**Nota**. Fuente: Gordillo Vázquez Francisco José, “Plasmas Fríos”, Revista Investigación y Ciencia, Junio 2008, 70-79.

Los plasmas se usan para corte de materiales, recubrimiento de superficies en la industria, esterilización en el área médica e incluso para el tratamiento de contaminantes.

## Sensores Y Tipos

Un sensor es un dispositivo que capta fenómenos que ocurren en el medio ambiente, el cual genera una señal eléctrica de salida que puede ser tratada para la obtención de información. Algunas de las aplicaciones de los sensores se encuentran reflejadas en la industria automotriz (detectar deslizamiento, la localización y orientación del pasajero, asistente de navegación, neumáticos inteligentes), aplicaciones médicas (química de análisis sanguíneos, diagnósticos de bajo costo y alta velocidad, dispositivos terapéuticos, implantes, test de ADN), aplicaciones militares (visión nocturna, navegación, sonar, radares), industria alimenticia, entre otras (Pallás, 2003).

En la tabla 2.2 se detallan todos los criterios de clasificación y ejemplos de cada clase de sensores.

Tabla 2. - Clasificaciones de sensores.

|  |  |  |
| --- | --- | --- |
| CRITERIO | CLASE | EJEMPLOS |
| Aporte de energía | Moduladores | Termistor |
| Generadores | Termopar |
| Señal de salida | Analógicos | Potenciómetros |
| Digitales | Codificadores de posición |
| Modo de operación | De deflexión | Acelerómetro de deflexión |
| De comparación | Servoacelerómetro |

**Nota**. Fuente: Pallás Areny Ramós, “Sensores y Acondicionadores de Señal”, 4ta ed., Barcelona, España, MARCOMBO, 2003, 358p.

## Gráficas Y Tipos

Una gráfica es un diagrama que emplea líneas, círculos, barras o alguna otra forma geométrica para representar datos tabulados (Martínez, 2006). Las gráficas se clasifican en cuatro grupos, los cuales se aprecian en la figura 2.1.

Figura 2. : Clasificación de las graficas

Fuente: Martínez Bencardino Ciro, “Estadísticas Básicas Aplicadas”, 3ra ed., Bogotá, Colombia, Ecoe Ediciones, 2006, 406p.

Diagramas

Superficiales

Lineales

De puntos

Rectilíneas

Curvilíneas

Circulares

Cuadradas

Triangulares

Rectangulares (Barras)

Piramidales

Prismáticas

Cúbicas

Estereométricas

Cartogramas

Cartodiagramas

Mapas Estadísticos

Pictogramas

GRÁFICAS

Esquemas de símbolos, objetos reales o figuras

Las gráficas como las tablas representan mucha información en poco espacio, pero por las siguientes ventajas se prefieren las gráficas por encima de las tablas, aunque se requiera de la existencia de una tabla para la creación de un a gráfica (Gutiérrez, 2005).

1. Mediante un gráfico es más fácil captar la atención, ya que un gráfico se aprecia la tendencia de los datos.
2. Es más fácil compara una gráfica con otra, que una tabla con otra tabla.
3. Las gráficas revelan de forma más rápida ciertos rangos que mediante un examen de una tabla no se podrían obtener fácilmente como son el valor máximo, valor mínimo, periodicidad, variaciones de pendiente, por mencionar algunas.
4. Mediante un gráfico es muy rápido y sencillo determinar algunas características de fenómenos.
5. Se emplea como ayuda visual en todo tipo de informes.
6. En el trabajo experimental las gráficas permiten encontrar la relación existente entre dos o más variables.

Las gráficas también tienen desventajas al presentar los datos experimentales, ya que no se pueden mostrar correctamente la precisión de las ediciones. Las gráficas se utilizan para obtener lectura entre los puntos experimentales mediante el proceso denominado interpolación. El proceso opuesto extrapolación consiste en prolongar la gráfica para obtener valores fuera del intervalo experimental (Gutiérrez, 2005).

## Android

Android es un sistema operativo móvil basado en Linux, creado inicialmente por la compañía de Android Inc., compañía que es adquirida en el año 2005 por Google. Siendo un sistema novedoso que trata de revolucionar la industria móvil día a día, ha logrado extenderse más allá de una plataforma móvil, para proveer de una plataforma de desarrollo para una gama más amplia de hardware, incluyendo Tablets y televisiones. Este sistema operativo se encuentra compuesto por (Satya & Dave, 2011):

1. Un documento de definición de compatibilidad (CDD) y un conjunto de pruebas de compatibilidad (CTS) que describen las capacidades requeridas por dispositivo poder soportar el software.
2. Un kernel Linux que provee de un bajo nivel de interfaz con el hardware, administrador de memoria y control de procesos, para una optimización total de los móviles y dispositivos integrados.
3. Librerías de código abierto para desarrollo de aplicaciones que incluye SQLite, WebKit, OpenGL y media Manager.
4. Un tiempo de ejecución que incluye que incluye Dalvik Virtual Machine (VM) y librerías del kernel que proveen de especificaciones de funcionalidad de Android, los tiempos de ejecución están diseñados para ser pequeños y eficientes para su uso en dispositivos móviles.
5. Un marco de aplicaciones que expone cada uno de los servicios del sistema para las capas de aplicación, incluyendo administrador de ventanas y gestor de localización, base de datos, telefonía y sensores.
6. Un kit de desarrollo de software (SDK) usado para crear aplicaciones, el cual está conformado de herramientas relacionales, plugins y documentación
7. Un conjunto de aplicaciones preinstaladas.

Los dispositivos móviles con Android típicamente vienen con una suite de aplicaciones preinstaladas que forman parte del proyecto de código abierto de Android (AOSP) algunas de las aplicaciones que lo integran son las siguientes (Reto, 2012).

1. Un cliente de e-mail.
2. Una aplicación de gestión de SMS (Short Message Services).
3. Una suite completa de PIM (Personal Information Management), conformada de un calendario y una lista de contactos.
4. Un navegador web basado en WebKit.
5. Una aplicación de grabación de cámara y video.
6. Una calculadora.
7. Una pantalla de inicio.
8. Un reloj de alarma.

Además los dispositivos móviles con Android también cuentan con las siguientes aplicaciones propietarias de Google Mobile (Satya & Dave, 2011).

1. La Google Play Store para descargar aplicaciones Android de terceros.
2. Una aplicación de Google Maps, que incluye vista de calles, indicador de viaje, turn-by-turn Navigation, imágenes de satélite y condiciones de tráfico.
3. El cliente de correo electrónico de G-mail.
4. El Google Talk de mensajería instantánea del cliente.
5. El reproductor de video de YouTube.

### Arquitectura De Android

El núcleo de la plataforma Android es un kernel Linux y es el responsable de los contadores de los dispositivos, acceso a los recursos, administrador o gestión de energía y otras tareas del sistema operativo. En el siguiente nivel, en la parte superior del kernel Linux están un numero de librerías de C/C++, tal como OpenGL, WebKit, FreeType, Secure Sockets Layer (SSL), librería de tiempo de ejecución de C (libc), SQLite y media frameworks, las cuales interactúan entre sí para lograr crear distintas aplicaciones.

Después de las librerías de C/C++ se encuentra el marco de aplicación. La mayoría del marco de aplicación accede al núcleo de la librería por medio de Dalvik. Dalvik es la máquina virtual y está optimizado para ejecutar múltiples instrucciones de la máquina virtual. Por último la API (Interfaz de Programación de Aplicaciones) para las bibliotecas principales de Android, incluye servicios de telefonía, recursos, lugares, interfaz de usuario (UI), proveedor de contenidos y gestión de paquetes (instalación, seguridad y así sucesivamente), algunos ejemplos de aplicaciones de usuario final son Home, contactos, teléfonos, navegador, entre otros (Reto, 2012). La figura 2.2 muestra la arquitectura por capas del sistema operativo Android.

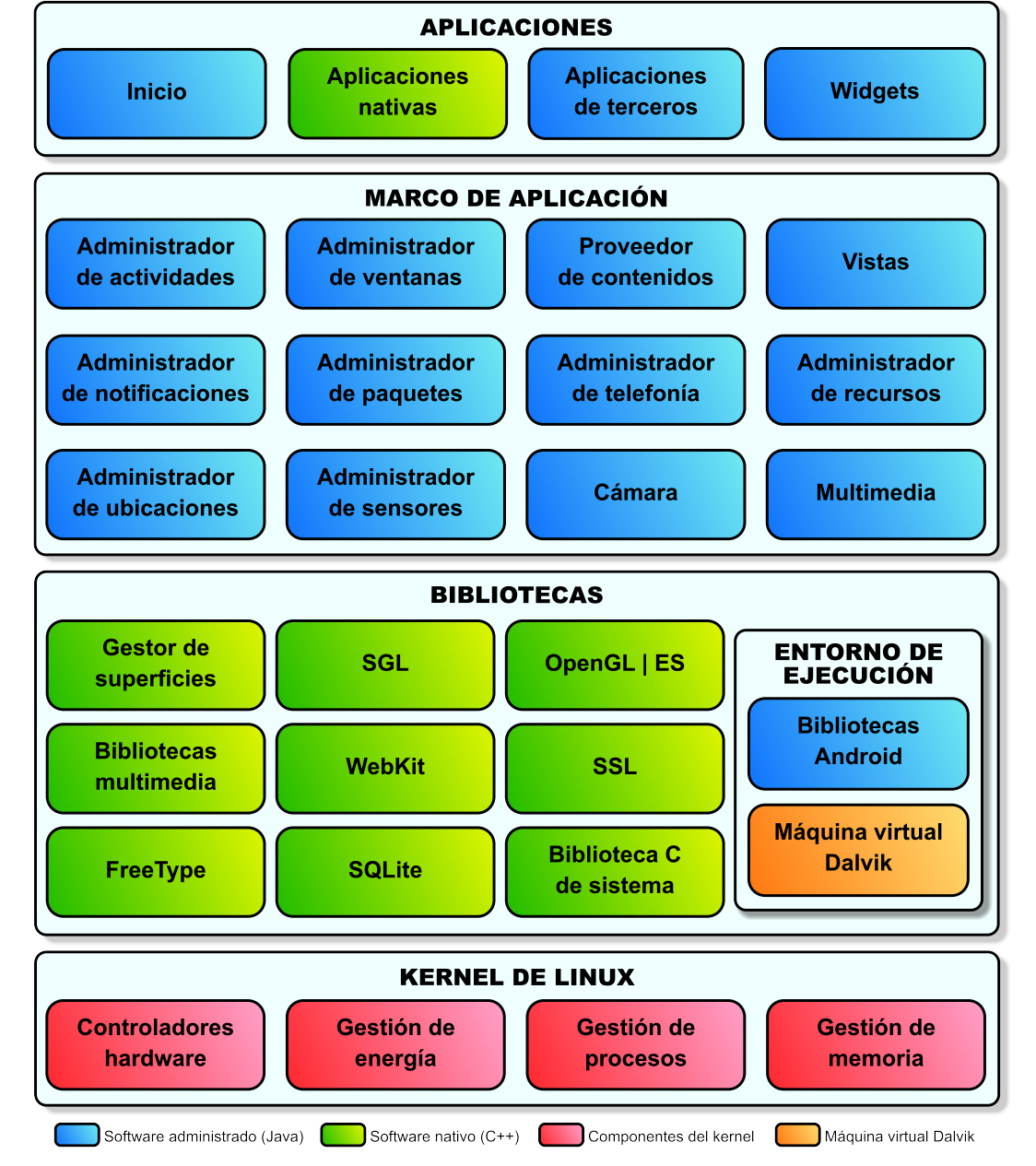


Figura 2. : Arquitectura del sistema operativo Android

Fuente: Reto Merier, “Profesional Android Aplication Development”, 1ra ed., Indiana, EEUU, 2012, 720p.

### Características De SDK De Android

1. GSM, EDGE, 3G, 4G y LTE son redes para telefonía o transferencia de datos que le permiten al usuario hacer o recibir llamadas o mensajes de SMS o para enviar o recuperar datos a través de las redes móviles.
2. APIs para servicios basados en localización tal como GPS y redes basadas en detección de ubicación.
3. Completo soporte para aplicaciones que se integran al mapa de control como parte de sus interfaces de usuarios hardware de acceso WiFi y conexiones punto a punto.
4. Control total de hardware multimedia, incluso grabación y reproducción con cámara y micrófono.
5. Librerías multimedia para reproducción, grabación de audio y video o imágenes en formato fijo.
6. API para el uso de sensores de hardware, tales como acelerómetro, brújula y barómetro.
7. Librerías para utilizar el bluetooth y hardware NFC (Near Field Comunication) para trasferencia de datos punto a punto.
8. IPC (Inter-Process Comunication) de pasos de mensaje.
9. Servicios en segundo plano, aplicaciones y procesos.
10. Un sistema integrado de código abierto HTML5 en un navegador web basado en WebKit.

## Software E Ingeniería Del Software

El software es el soporte lógico de un sistema informático, el cual hace posible la realización de tareas específicas, dentro de la ingeniería del software abarcaría no solo a los programas sino a todos los documentos asociados, además de la configuración de datos que se necesitan para hacer que estos programas operan de manera correcta (Sommerville, 2005); (ver Anexo 1).

### Ingeniería Del Software

Se define a la ingeniería del software como la aplicación de un enfoque sistemático, disciplinado y confiable al desarrollo, operación y mantenimiento del software, siendo el establecimiento y uso de principios fundamentales de la ingeniería con el objetivo de desarrollar de una forma económica software que sea de alta calidad, facilidad de mantenimiento y con un entendimiento total del problema antes del desarrollo (Naur, 1969; IEEE, 1990).

La ingeniería de software completa el proceso del software, definiendo un proceso como un enfoque adaptable que el equipo de software busque y elija el conjunto de apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes lo usan. La estructura de procesos incluye una serie de actividades que son adaptables a través de todo proceso del software, la ingeniería del software consta de cinco actividades comunicación, planeación, modelado, construcción y despliegue (S. Pressman, 2010), descritas cada una en el Anexo 2.

Dentro del escrito de McConnell (1999) afirma que las actividades estructurales del proceso de ingeniería de software son complementos y se centran en la administración, el seguimiento y el control del proyecto como son avances, la calidad, el cambio y los riesgos (ver Anexo 3).

## Metodología Para El Desarrollo De Software

Una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto, hasta que cumplimos el objetivo por el cual fue creado (Loïc & Segovia, 2005; Benet, 2003).

### Clasificación De Las Metodologías

1. **Metodología estructurada**: la orientación de esta metodología se dirige hacia los procesos que intervienen en el sistema a desarrollar, es decir, cada función a realizar por el sistema se descompone en pequeños módulos individuales. Es más fácil resolver problemas pequeños, y luego unir cada una de las soluciones, que abordar un problema grande.
2. **Metodología orientada a objetos**: a diferencia de la metodología mencionada anteriormente, ésta no comprende los procesos como funciones sino que arma módulos basados en componentes, es decir, cada componente es independiente del otro. Esto nos permite que el código sea reutilizable. Es más fácil de mantener porque los cambios están localizados en cada uno de estos componentes.
3. **Metodologías formales**: a diferencia de las metodologías mencionadas anteriormente esta parte de especificar las necesidades de información en base a un modelo puramente matemático, demostrando matemáticamente que un programa es correcto en el sentido de que se ajusta a las necesidades. Su utilización en el desarrollo de software es poco frecuente en la actualidad, a causa de la complejidad un modelado tan detallado y formalizado en casos reales.

## Ciclo De Vida Del Software

La producción de software es algo que más que la programación, hay etapas que la preceden y otras que la siguen. El ciclo de vida del software está constituido por el conjunto de todas estas etapas. Los métodos y técnicas de la ingeniería en software están delimitados por el ciclo de vida del software y más concretamente, por las diferentes etapas que se conforman. Existen distintos modelos del ciclo de vida del software por esta razón no hay ninguno que sea ideal o que no tenga grandes limitaciones (Tuya & Ramos, 2007; Campderrich, 2003). En la figura 2.3 se aprecian los procesos que constituyen al ciclo de vida del software antes explicado, (ver Anexo 4).

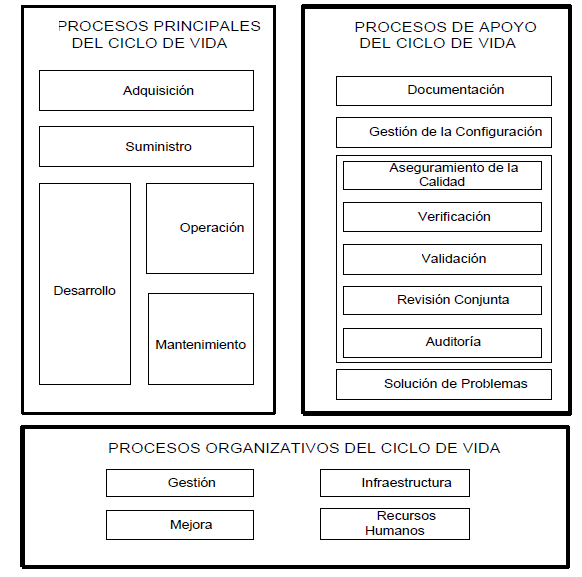


Figura 2. : Diagrama de procesos del ciclo de vida del software

Fuente: Loïc Martínez Alonso Fernando, Segovia Javier Fco., "Introducción a la Ingeniería del software: Modelos de desarrollo de programación", 1ra ed., Madrid, España, Delta Publicaciones, 2005, 538p.

## Modelos De Proceso

Existen diversos modelos, paradigmas y filosofías del ciclo de vida del software, en los cuales se apoya para la construcción del software. En la tabla 2.3 se muestran algunos de los modelos de desarrollo de software más conocidos.

Tabla 2. - Modelos de proceso del ciclo de vida del software.

|  |  |  |  |
| --- | --- | --- | --- |
| **MODELOS DE PROCESO** | | | |
| Modelos Perspectivos | Modelos Agiles | Modelos Genéricos | Modelos Específicos |
| Modelo en cascada | Programación Extrema (XP) | Capability Maturity Model (CMM) | Unified Process UP |
| Modelo en V | Crystal Methodologies | Capability Maturity Model Integration (CMMI) | Rational Unified Process RUP |
| Modelo iterativo | Rapid Application Development (RAD) | ISO 9001-2000 Sistema para administración de la calidad | Personal Software Process PSP |
| Modelo de desarrollo incremental | Dynamic Systems Development Method (DSDM) | ISO/IEC 15504 Marco para evaluación de procesos de software. | Team Software Process TSP |
| Modelo en espiral | Adaptive Software Development (ASD) |  |
|  | Agile Unified Process (AUP) | MoProSoft Modelo de procesos para la industria de software en México |
| Feature -Driven Development (FDD) |  |
| Lean Software Development (LSD) |
| Essential Unified Process (EssUP) |
| Open Unified Process (OpenUP) |
| Kanban |
| SCRUM |

**Nota**. Fuente: Sommerville Ian, “Ingeniería en Software”, 7ma ed., Madrid, España, 2005, 712p. López Lirra Roberto, “Modelos específicos”, Revista Software Guru Conocimiento En Práctica, Febrero 2005, no 1. McGlaughlin R, “Software Engineering Notes”, vol. 16, núm. 4, 1991, 53-54 pp.

## Proceso Personal De Software PSP

Personal Software Process (PSP) es un proceso de auto mejora que ayuda a controlar, administrar y mejora la forma de trabajar, lo que lo hace un marco estructurado de orientación y procesos para el desarrollo de software. PSP está basado en “*La calidad de software depende del trabajo de cada uno de los ingenieros de software*”. Debido a que los costos de personal constituyen 70% del costo del desarrollo de software, las capacidades y hábitos de trabajo de los ingenieros determinan en gran manera los resultados del desarrollo de software (Humphrey, 2005).

Basado en prácticas encontradas en CMM, el PSP puede ser usado por ingenieros para estructurar y disciplinar el desarrollo de software. El ingeniero de software podrá planear mejor el trabajo, conocer con precisión el desempeño, medir la calidad de productos, y mejorar las técnicas. En la figura 2.4 se denotan las fases que constituyen al proceso personal de software. De acuerdo con el artículo “*Procesos de Software*” de Rubalcaba (2005) el PSP puede ser aplicado en:

* 1. Desarrollo de programas.
  2. Definición de requerimientos.
  3. Documentación.
  4. Pruebas de sistemas.
  5. Mantenimiento de sistemas.

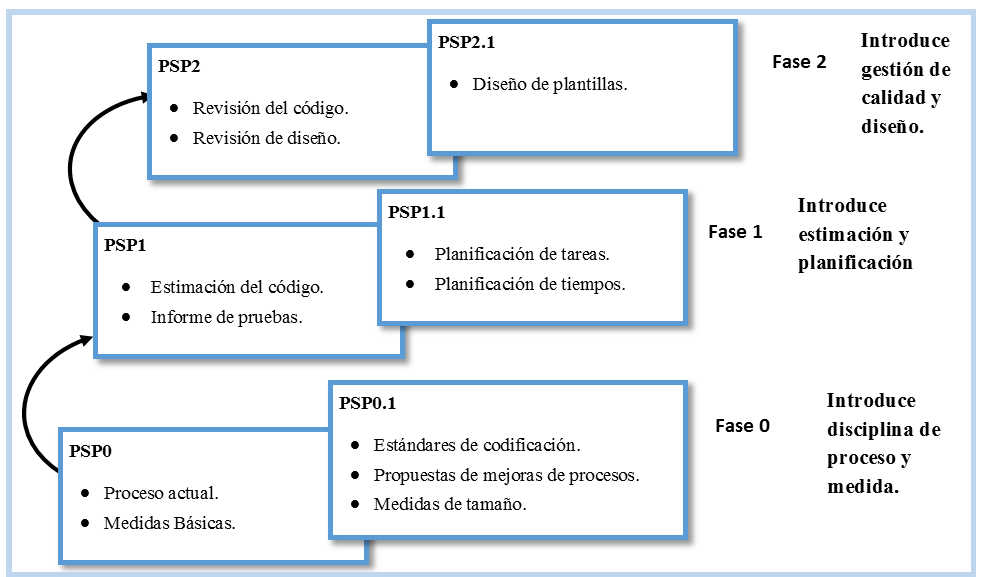


Figura 2. 4: Diagrama de las etapas del proceso personal de software.

Fuente: Humphrey S. Watts, “PSP: A Self-Improvement Process for Software Engineers”, 1ra ed., Massachusetts (EEUU), Person Education, 2005, 210 p.

### Fases Del PSP

Fase 0: El objetivo principal del ingeniero de software es aprender a seguir un proceso definido y a recopilar los datos básico sobre tamaño, tiempo y defectos.

Fase 1: el objetivo principal de es la estimación y la planeación. Para ello se requiere de conocer métodos o estándares de estimación de esfuerzo y de tamaño, y emplear los resultados obtenidos para la planificación y el seguimiento de proyecto.

Fase 2: debido a que se realizó previamente el control de la planeación el objetivo principal está en la gestión de calidad, para ello es necesario de conocer los métodos de detección temprana y eliminación de defectos.

## Lenguaje De Modelado Unificado

UML o Lenguaje de Modelado Unificado (Unified Modeling Language) es sucesor de la oleada de métodos y análisis y diseño orientado a objetos (OOA&D) que surgió a finales de ña década de 1980 y principios de la siguiente. El lenguaje de modelado es la notación gráfica de que se valen los métodos para expresar los diseños (Fowler Martin & Scott, 1999).

## Sistema De Información

Un sistema de información es un conjunto de elementos o componentes interrelacionados para recolectar (entradas), manipular (procesó), y diseminar (salida) datos en información y proveer de un mecanismo de retroalimentación en pro del cumplimiento de un objetivo (Stair & Reynolds, 1999), en la figura 2.5 se aprecia los componentes que integran a un sistema de información.

1. Entrada: es la actividad que consiste en recopilar y capturar los datos primarios, las cuales son un proceso manual o automatizado.
2. Procesamiento: es la conversión y transformación de los datos en salidas útiles, esto implica cálculos, realizar comparaciones y adoptar acciones alternas, y el almacenamiento de datos para el uso posterior. El procesamiento se puede realizar de manera manual o por medio de una computadora.
3. Salida: implica producir información útil, por lo general en forma de documentos o reportes, en algunos casos la salida de un sistema podría ser la entrada de otro.
4. Retroalimentación: es la salida que se emplea para para afectar cambios en actividades de entrada o procesamiento. Es de gran importancia para administradores y tomadores de decisiones, es decisiva para la exitosa operación de un sistema.

**Sistema de Información**

**Sistema de Información**

**Retroalimentación**

**Retroalimentación**

**Procesamiento**

**Procesamiento**

**Entrada**

**Entrada**

**Salida**

**Salida**

Figura 2. : Componentes de un sistema de información.

Fuente: Stair M. Ralph, Reynolds W. George, “Sistemas de Información: Enfoque Administrativo”, 4ta ed., DF (México), Internacional Thomson Editoriales, 1999, 700p.

Figura 2. : Componentes de un sistema de información.

Fuente: Stair M. Ralph, Reynolds W. George, “Sistemas de Información: Enfoque Administrativo”, 4ta ed., DF (México), Internacional Thomson Editoriales, 1999, 700p.

## Base De Datos

Una base de datos es una estructura de computadores integrada, compartida que aloja un conjunto de datos para el usuario final, almacenados en conjunto sin redundancias innecesarias; cuya finalidad es la de servir a una o más aplicaciones de manera eficiente. En cierta forma una base de datos se parece a un archivero electrónico bien organizado en el que un poderoso software, conocido como sistema administrador de base de datos ayuda en el manejo de su contenido (Rob & Coronel, 2004).

Las bases de datos permiten mejorar la calidad de las prestaciones de los sistemas de información y aumentar su rendimiento. De acuerdo con Nevado (2006) las bases de datos tienen las siguientes ventajas.

* Independencia de los datos y los programas y procesos.
* Menor redundancia.
* Integridad de los datos.
* Coherencia de los resultados.
* Mayor seguridad de los datos.
* Datos más documentos.
* Acceso a los datos más eficientes.
* Reducción de espacio de almacenamiento.
* Acceso simultaneo a los datos.

Las bases de datos también tienen una serie de desventajas:

* Instalación costosa.
* Larga y difícil puesta en marcha.
* Rentabilidad a mediano plazo.
* Personal especializado.
* Complejidad en la recuperación de datos.
* Tamaño.
* Complejidad.

### Sistema Administrado De Bases De Datos

Un sistema administrador de base de datos, por sus siglas en ingles DBMS “Data Base Management System” es un conjunto de programas que maneja la estructura de la base de datos y controla el acceso a los datos guardados en ésta. El DBMS permite compartir los datos de la base de datos entre múltiples aplicaciones y usuarios (Rob & Coronel, 2004; Nevado, 2006).

El DBMS actúa como intermediario entre el usuario y la base de datos, ya que transforma las solicitudes de usuario en código complejo necesario para atender dichas solicitudes. El DBMS guarda mucha de la complejidad interna de la base de datos, de los programas de aplicación que se ésta empleando (Nevado, 2006).

El éxito de DBMS reside en mantener la seguridad e integridad de los datos. Proporciona una serie de herramientas a los distintos usuarios, entre las herramientas que proporciona están:

* 1. Herramienta para la creación y especificación de los datos.
  2. Herramienta para la administrar y crear la estructura física.
  3. Herramienta para la manipulación de datos.
  4. Herramienta de recuperación en caso de desastre.
  5. Herramienta para la creación de copias de seguridad.
  6. Herramienta para la gestión de la comunicación de la DB.
  7. Herramienta para la creación de aplicaciones.
  8. Herramienta de instalación de la DB.
  9. Herramienta para la exportación e importación de datos.

## XLM

XML por sus siglas en ingles eXtensible markup Languaje (lenguajes de marcas extensibles), es un metalenguaje extensible de etiquetas por el World Wide Web Consortium (W3C). Por lo tanto XML no es un lenguaje como tal, sino un servicio por el cual se pueden realizar aplicaciones modulares auto-descriptivas que se publican, se invocan y se ubican desde cualquier lugar Web (Patrick, 2002).

Es una y tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tienen un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

## Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura (Deitel & Deitel, 2010).

Java le permite jugar en línea, participar en sesiones de chat con internautas de todo el mundo, calcular los intereses de una hipoteca y ver imágenes en tres dimensiones, entre otras muchas aplicaciones. Es también esencial para las aplicaciones de intranet y otras soluciones de comercio electrónico que constituyen la base informática de las empresas. Su versatilidad, eficiencia y portabilidad, Java se ha convertido en un recurso inestimable ya que permite a los desarrolladores (Gosling, Joy e Steeles, 2000).

1. Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma.
2. Crear programas para que funcionen en un navegador web y en servicios web.
3. Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
4. Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
5. Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier tipo de dispositivo digital.

# Capítulo III - Análisis De La Aplicación

## Análisis Del Sistema

Objetivo principal de esta actividad es realizar un análisis global del sistema, estableciendo los requisitos de todos los elementos del sistema, contemplando las tareas manual e informáticas que lo definen.

## Ingeniería De Requerimientos

La ingeniería de requerimientos es el espectro amplio de tareas y técnicas para describir, documentar y mantener un conjunto de requerimientos. Incluye siete tareas diferentes concepción, indagación, elaboración, negociación, especificación, validación y administración, algunas de estas tareas ocurren de manera paralela y todas se adaptan a la necesidad del proyecto (ver Anexo 5).

## IEEE 830 Practica Recomendada Para Requisitos De Especificaciones De Software

El IEEE 830 es un documento patrocinado por el comité de estándares de ingeniería de software de la sociedad de computación del IEEE el cual fue aprobado el 25 de junio de 1998 por la IEEE SA Standards Bard.

Describe los criterios recomendados para la especificación de requerimientos de software dentro de cinco clausulas.

* Clausula 1: explica el alcance de esta práctica recomendada.
* Clausula 2: es un listado de referencias realizadas para otros estándares.
* Clausula 3: provee las definiciones de los términos utilizados.
* Clausula 4: provee de información para lograr la escritura de un plantilla de especificación de requisitos de software (Software requirements specification o SRS).
* Clausula 5: se analiza cada una de las partes que integran a un SRS.

Este documento contiene dos anexos uno que proporciona plantillas de formatos alternos, que establece los lineamientos para el cumplimiento con los estándares IEEE/EIA 12207.1-1997 (ver Anexo 6).

### Propósito Del SRS

Mediante este documento pretende establecer un SRS apegándose lo más posible a la norma IEEE 830. SQLGraphic es una aplicación nativa en el sistema operativo Android, para el manejo y graficación de datos adquiridos de una simulación de plasma no térmico, esta herramienta ayudara a la interpretación de la información, además de tener disponible la información en cualquier momento, situación y lugar. Entendiendo la importancia del análisis de resultados y el uso de gráfica de toda investigación experimental, se pretende establecer una definición compleja y global de las operaciones y funciones de la aplicación SQLGraphic con el fin de establecer los requerimientos plateados por el usuario.

### Alcance Del SRS

Crear gráficas generadas a partir de datos extraídos de una base de datos externa a la aplicación móvil, usando sentencias SQL. El gestor de base de datos podrá ser MySQL, SQLite o algún otro, será necesario que el usuario se autentifique para acceder a la información.

Por medio del uso de sentencias SQL el usuario podrá seleccionar las tablas, campos, realizar toda aquellas funciones que soporte el gestor de la base de datos, se usa dos tipos de sentencia para mostrar los resultados, la primera tiene como consecuencia un tabla, mientras que la segunda genera una gráfica de los datos.

### Personal Involucrado

En el tabla 3.1 y 3.2 se describen cada uno de los roles que desempeñara las personas involucradas en el proyecto de la aplicación móvil.

Tabla 3. - Analista de Diseño Junior.

|  |  |
| --- | --- |
| Nombre | Rafael Samano Pichardo |
| Rol | Analista de Diseño, Desarrollador y Tester Junior |
| Categoría Profesional | Pasante en Ingeniería en Computación |
| Responsabilidad | Persona comisionada para realizar el análisis y diseño del sistema utilizando diversas herramientas como son platilla SRS, COCOMO II y UML, además de crear la documentación necesaria que abordara dichas tareas. Por ultimo deberá administrar la arquitectura, el proceso de desarrollo e implementación del sistema en el dispositivo móvil, para realizar las pruebas necesarias para su la corrección, en caso de ser necesarias. |
| Información de contacto | Teléfono Cel.: 7121403166  Correo Electrónico: rafasp020689@gmail.com |
| Aprobación | Dr. Carlos Torres Reyes |

Tabla 3. - Cliente de la aplicación.

|  |  |
| --- | --- |
| Nombre | Dr. Carlos Torres Reyes |
| Rol | Cliente de la aplicación |
| Categoría Profesional | Doctor en Electrónica |
| Responsabilidad | El Cliente es la persona que se encargara de la revisión de la aplicación de software para el sistema operativo Android, dentro de sus tareas esta verificar que la aplicación cumpla con los requerimientos documentados, al igual, será la encargada de dar el grado de satisfacción en cuanto a la realización de la aplicación. Buscará errores funcionales, de presentación o de incumplimientos de algún tipo de requerimiento. |
| Información de contacto | Teléfono Cel.: 7223876234  Correo electrónico: edtore75@hotmail.com |

### Perspectiva Del Producto

La aplicación SQLGraphic para el sistema operativo móvil Android, interactuara con dispositivos móviles como Smartphone y Tabletas, la aplicación será compatible con versiones posteriores a la 4.0 denominada como Ice Cream Sandwich. La interfaz del usuario será una interfaz con los estándares de diseño de Android, evitando errores comunes como es el sobre-diseño, sub-diseño, sobrecarga de diseño, por nombrar algunas.

### Funcionalidad Del Producto

La aplicación contemplara dentro de su estructura las siguientes funciones.

1. Conexión con el Sistema gestor de base de datos (DBMS): su función radica en establecer una conexión directa con sistemas de gestión de bases de datos de forma totalmente online. Para ella se deberá crear un registro público en el dispositivo móvil con los datos del servidor externo.
2. Selección de base de datos: después de tener acceso al DBMS se lograra realizar esta función a través de un botón de selección, el cual desplegara en una lista de mensaje con las base de datos que integren al DBMS. Además de que el usuario tenga la versatilidad de interactuar con las demás bases de datos.
3. Sentencia SQL: se encarga de capturar la redacción de una sentencia SQL, para lograr una versatilidad total del manejo de los datos, soportando todas aquellas funciones del gestor de base de datos. Pero excluirá aquellas funciones que tengan un impacto sobre la base de datos.
4. Sentencia de Graficación: se apoyara de la sentencia SQL para no tener la necesidad de crear una sentencia única para graficar, para poder discrepar entre una y otra se utilizara un identificador.
5. Graficación por relación de correspondencia entre variables: se creara la gráfica en base a coordenadas lineales generadas a partir de los campos seleccionados con respecto al campo base, interpretando la sentencia de sentencia SQL.

### Características Del Usuario

En la tabla 3.3 se describen a detalle las características que debe tener el usuario para que sea posible utilizar correctamente la aplicación móvil.

Tabla 3. - Características del usuario final.

|  |  |
| --- | --- |
| Tipo de Usuario | Administrador |
| Formación | Ingeniería, licenciatura en sistemas computacionales o carrera afín. |
| Habilidades | Contar con conocimientos de matemáticas, electrónica y computación, manejo de base de datos y lenguaje SQL, para una comprensión de la interface de usuario y de las gráficas desplegadas por el sistema móvil. |
| Actividades | Algunas de las actividades a destacar será la selección de la base de datos, seleccionar el tipo de sentencia y creación de sentencias SQL. |

### Restricciones

Actualmente la tendencia en dispositivos móviles se inclinan a pantallas cada vez más amplias, esto significa que un Smartphone con pantalla inferior a 4 pulgadas puede parecer obsoleto, pero solo significa que el diseño de la interfaz se tiene que delimitar a un área reducida.

A la hora de diseñar se deben de tomar en cuenta algunos errores comunes como son de funcionalidad, usabilidad, sobre-diseñar, sub-diseñar, diseños exageradamente pequeños, sobrecarga de diseño, entre otros. El costo que tendrá el servidor para realizar las operaciones de estructuración, delimitación de campos y creación lineal de las gráficas, al utilizar la sentencia SQL.

El requerimientos de hardware pueden ser estrictos a la hora de ajustarse al sistema operativo, para ello es necesario tener en cuenta como mínimo las siguientes características para poder correr un aplicación en cualquier dispositivo móvil con sistema operativo Android. Procesador con velocidad mínima de 650 MHz, memoria RAM de 512 ya que el SO ocupa un mínimo de 60 Mb y si se contempla que se manejan aplicaciones en segundo plano, entonces quedan pocos megabytes para abrir y gestionar otras aplicaciones, memoria interna mínimo 750 Mb, batería no menor a los estándares actuales (Matt & Gary, 2006).

### Suposiciones Y Dependencias

Dentro de los factores que pueden alterar al sistema se cuentan la perdida de conexión en algún momento de acceso a la información. Otro factor seria agregar funciones nuevas a la estructura original del diseño de la aplicación.

El uso de un lenguaje de programación diferente a Java y XML para la realización de requerimientos futuros. Modificación de alguna función específica del sistema, en base al diseño y a la presentación de las gráficas.

### Evolución Previsible Del Sistema

Los requisitos que se plantean a futuro son el manejo de información sin necesidad de tener acceso directo a la base de datos, siendo considerado como una razón de vulnerabilidad para el usuario, aun cuando se cifre los datos, crear un diseñador automático de sentencias SQL o procesamiento de archivos de Excel.

Poder modificar, añadir y eliminar datos directamente en la gráfica y el resultado de dichas acciones se vean reflejadas directamente en la o las tablas de la base de datos. Además de migrar el sistema a un web app ofrecería la ventaja de que los procesos matemáticos se le delegarían totalmente al server. Con este tipo de sistema se puede manejar herramientas como Node.js, Ruby, Django, CSS3, HTML5 entre otras, logrando tener una interface muy semejante en a la que hay en el mercado de los dispositivos móviles.

Poder migrar la aplicación a otros sistemas operativos móviles como Windows Phone, iOS, lograría un mercado mucho mayor y englobando totalidad de los usuarios móviles.

### Requisitos Específicos

De las tablas de la 3.4 a 3.8 se describen cada uno de los requisitos específicos o restricciones de la aplicación, además de la prioridad que se le asigna a cada uno de ellos, delimitar estos requisitos ayuda a saber que podrá o no hacer el usuario dentro del sistema.

Tabla 3. - Requisito 1: Acceso a la información.

|  |  |
| --- | --- |
| Número de requisito | 1 |
| Nombre de requisito | Acceso a la información |
| Tipo | Requisito |
| Fuente de requisito | Se requiere que el usuario como tal sea capaz de seleccionar los datos alojados en una base de datos a placer.  Las bases de datos requieren que la persona que busca acceder a la información alojada en ella, se autentifique con sus credenciales de usuario.  El usuario deberá crear un registro local en el dispositivo móvil el cual contenga los datos del sistema gestor de base de datos, como son: nombre de asignación local, host, puerto, nombre de usuario, contraseña y base de datos si es necesario. |
| Prioridad | Alta |

Tabla 3. - Requisito 2: Sentencia SQL.

|  |  |
| --- | --- |
| Número de requisito | 2 |
| Nombre de requisito | Sentencia SQL |
| Tipo | Requisito |
| Fuente de requisito | Es requerido que el sistema acepte sentencias SQL para poder ofrecer una alternativa más al usuario que solo la graficación de los datos, además de proporcionar auxiliares de creación de sentencias SQL. Se pueden guardar sentencias SQL localmente para proporcionar el acceso a ellas. |
| Prioridad | Alta |

Tabla 3. - Requisito 3: Sentencia de graficación de datos.

|  |  |
| --- | --- |
| Número de requisito | 3 |
| Nombre de requisito | Sentencia de graficación de datos |
| Tipo | Requisito |
| Fuente de requisito | Se usa una sentencia SQL para la graficación de datos para no tener que crear una sentencia específica para esta función particular. |
| Prioridad | Alta |

Tabla 3. - Requisito 5: Exposición de los datos en las gráficas.

|  |  |
| --- | --- |
| Número de requisito | 5 |
| Nombre de requisito | Exposición de los datos en las graficas |
| Tipo | Requisito |
| Fuente de requisito | El diseño de la gráfica deberá permitir una la comprensión correcta de los datos, aun cuando los datos tengan un valor de millones o de milésimas. Al mostrar la gráfica se podrá guardar una imagen el sistema de ficheros del dispositivo móvil. |
| Prioridad | Alta |

## Requisitos Comunes De Las Interfaces

### Interfaces De Usuario

* **Interface de inicio**: se conformara de la barra de acción, compuesta por el logo de la aplicación y un botón con el icono de nuevo, el cual tendrá la función de agregar los datos del sistema gestor de base de datos. La parte central estará conformada por una lista de los servidores registrados localmente por el usuario, los colores que se eligieron para la aplicación están fundados completamente en la psicología del color y contemplado los estándares de diseño de Android.
* **Interfaz para agregar información**: estará estructurada por la barra acción formada por el logo del sistema de la aplicación, un icono para guardar la información del gestor y el icono de test de conexión. La parte central contra los campos de texto con el nombre local del gestor de base de datos, el hosting, el puerto que por default 3306, nombre de usuario, la contraseña y la base de datos.
* **Interfaz de trabajo**: se ejemplifica de acuerdo a estándares de Android en diseño de aplicaciones móviles, conformada por la barra de acción, que a su vez la integraran el icono de la aplicación, un icono para guardar sentencias SQL, otro icono que ejecutara la sentencia de tipo tabla o de graficación en el sistema gestor de base de datos. La siguiente sección de la interfaz estar compuesta por un spinner para seleccionar la base de datos, una pareja de radio botones para definir el tipo de sentencia a ejecutar. Para la capturar los dos tipos de sentencia se utilizara un campo de texto, la parte final estará conformada por 3 pestañas de palabras clave, operadores y tablas que integran una interfaz común de conexión remota.
* **Interfaz de ejecución SQL:** conformada por la barra de acción, la cual estará integrada por el logo y por un icono de menú, la parte central la ocupara una tabla dinámica de columnas y filas variables de acuerdo a la sentencia SQL, al pie de la interfaz tendrá una barrar de estado con el número de páginas que llegue a generar la sentencia.
* **Interfaz de graficación:** Estáintegrada por la gráfica generada a partir de la sentencia de tipo graficación, además de la barra de acción completada por el logo y el icono que hace referencia a guardar, el cual genera una imagen en el sistema de ficheros del dispositivo móvil.

### Interface De Hardware

Dispositivo móvil (Smartphone, tabletas, etc.) con Ice Cream Sandwich instalado o cualquier versión superior a la 4.0, además de contar con una conexión a internet a través de Wi-Fi o algún plan de conexión móvil adquirido mediante un proveedor de telefonía móvil.

### Interface De Comunicación

Conexión directa con el sistema gestor de base de datos a través del protocolo TCP/IP, RDA en caso de que el dispositivo móvil cuente con conexión Wi-Fi. De modo contrario se empleara la tecnología 3G, para la transición de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles).

## Requisitos Funcionales

### Requisito Funcional Número Uno

|  |  |
| --- | --- |
| Id requerimiento | 1 |
| Descripción | Generar el registro local de con los datos del servidor. |
| Entradas | Datos de acceso obligatorios: nombre local, hosting, puerto definido en 3306, nombre del usuario. |
| Salidas | Se admitirá un registro local en la base de datos del sistema móvil, con los datos requeridos para hacer la conexión con el servidor remoto. |
| Proceso | Se agregan los datos en cada uno de los campos de texto dentro de la actividad, se realizar el testeo de la conexión y se prosigue a guardar los datos en la base de datos local. |
| Situación anormal | Si por alguna situación las credenciales de acceso son incorrectas o el tiempo de acceso se agota, el sistema de la aplicación envía un mensaje de fallo en la conexión con el servidor externo. |
| Precondición | El servidor debe permanecer online mientras se accede a él.  Los datos del sistema gestor de base de datos deben de encontrarse añadidos en el formulario.  Mientras que dispositivo móvil debe tener algún tipo de conexión con la nube, ya sea por un plan tarifario o por Wi-Fi. |
| Post-Condición | El servidor de la aplicación como de la base de datos remota deben permanecer online durante la conexión para la obtención de la información. |
| Efecto colateral | Se crea una conexión directa con el sistema gestor de base de datos y posteriormente se agregar un registro a la base local de la aplicación. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Dos

|  |  |
| --- | --- |
| Id requerimiento | 2 |
| Descripción | Permitirá conexión al sistema gestor de base de datos del servidor, haciendo uso de las credenciales de autenticación. |
| Entradas | Datos de acceso obligatorios: hosting, puerto definido en 3306, nombre del usuario, contraseña. |
| Salidas | Confirmación las credenciales en el servidor, hacer uso de la información que integra al servidor como base de datos, tablas y columnas, mediante un mensaje de alerta se informa si la conexión es exitosa o fallo. |
| Proceso | Si la acción se realiza en la actividad de agregar, se obtienen los datos para la conexión de los campos agregados por el usuario, en caso contrario se adquieren de la base local, posteriormente se intenta conectar con el servidor externo. Si la acción tiene éxito el usuario puede tener acceso a la información del servidor. |
| Situación anormal | Si alguno de los campos que conforman el registro local de la base de datos no es correcto. El tiempo de espera para acceder al servidor se agota. El sistema gestor de base de datos tiene algunas funciones restringidas y no se puede tener acceso a la información del mismo. |
| Precondición | El servidor debe permanecer online mientras se accede a él.  Los datos del sistema gestor de base de datos deben de encontrarse añadidos en el formulario o alojados en la base local.  El dispositivo móvil debe tener algún tipo de conexión con la nube, ya sea por un plan tarifario o por Wi-Fi. |
| Post-Condición | Los servidores tanto de la aplicación como de la base de datos remota deben permanecer online durante la conexión para obtener los datos. |
| Efecto colateral | Se obtienen los datos del sistema gestor de base de datos. Se puede ejecutar sentencias en la base de datos seleccionada. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Tres

|  |  |
| --- | --- |
| Id requerimiento | 3 |
| Descripción | Permitirá seleccionar una base de datos que están alojadas al sistema gestor de base de datos. |
| Entradas | Datos de acceso obligatorios: hosting, puerto definido en 3306, nombre del usuario, contraseña. |
| Salidas | Realizar la selección de base de datos y del tipo de sentencia que el usuario pretende usar. |
| Proceso | Cuando la el sistema de la aplicación se conecta remotamente a sistema gestor de base de datos adquiere toda la información que integra al administrador como base de datos y tablas. Consecutivamente las admite dentro de un objeto para que el usuario pueda manipular dicha información. |
| Situación anormal | Que el tiempo de petición al servidor concluya o que se pierda la conexión con el servidor.  Que el servidor tenga algún tipo de restricción sobre la información que se solicita. |
| Precondición | El servidor permanezca online mientras se accede a él. Los datos del sistema gestor de base de datos se encuentren alojados en la base local. El dispositivo móvil debe tener algún tipo de conexión con la nube, ya sea por un plan tarifario o por Wi-Fi. |
| Post-Condición | El servidor remoto y el de la aplicación permanezcan online. |
| Efecto colateral | Se aloja la información adquirida del sistema gestor en objetos para facilitar la manipulación de los datos. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Cuatro

|  |  |
| --- | --- |
| Id requerimiento | 4 |
| Descripción | Seleccionar el tipo de sentencia a realizar sea de tipo tabla o de graficación. |
| Entradas | Datos de acceso obligatorios: id del servidor alojado, base de datos seleccionada. |
| Salidas | Habilitar el campo de texto en donde se captura la sentencia. |
| Proceso | Manipular el objeto que corresponda al tipo de sentencia que el usuario procesara. El objeto antes mencionado debe diferenciar entre los dos tipos de sentencias. |
| Situación anormal | Que la actividad no se cargue de forma adecuada la información que integra la gestor de base de datos, esto imposibilita el uso de la aplicación. |
| Precondición | La información para autentificarse en el sistema gestor de base de datos se encuentre alojada en la base local de la aplicación. |
| Post-Condición | Generar una estructura que permita identificar el tipo de sentencia a manipular. |
| Efecto colateral | Se habilita el objeto en donde se ingresar la sentencia. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Cinco

|  |  |
| --- | --- |
| Id requerimiento | 5 |
| Descripción | Procesar sentencia SQL o de graficación. |
| Entradas | Datos de acceso obligatorios: id del servidor alojado, base de datos seleccionada y tipo de sentencia. |
| Salidas | La sentencia de tipo tabla crea una tabla que dependerá únicamente de la sintaxis de la sentencia, en contra parte la sentencia de graficación genera un grafica en base a un campo base. |
| Proceso | Inmediatamente después seleccionar la base de datos, el tipo de sentencia e ingresar la sentencia, se ejecuta la sentencia utilizando un objeto botón, el cual realiza una discrepancia de acuerdo al tipo de sentencia y ejecuta la actividad correspondiente a la sentencia. |
| Situación anormal | Algún error interno que no permita que se ejecute la actividad correspondiente al tipo de sentencia o el envió de la información a la nueva actividad. |
| Precondición | Seleccionar de la base de datos, redactar la sentencia SQL. |
| Post-Condición | El servidor permanezca online mientras se accede a él. Los datos del sistema gestor de base de datos se encuentren alojados en la base local. El dispositivo móvil debe tener algún tipo de conexión con la nube, ya sea por un plan tarifario o por Wi-Fi. |
| Efecto colateral | Se iniciara una actividad de acuerdo al tipo de sentencia que se haya seleccionado. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Seis

|  |  |
| --- | --- |
| Id requerimiento | 6 |
| Descripción | Crear actividad para procesar sentencia de tabla |
| Entradas | Datos de acceso obligatorios: id del servidor alojado, base de datos seleccionada y sentencia. |
| Salidas | Crear una tabla basada en la sentencia que el usuario haya planteado de acuerdo a sus necesidades actuales. |
| Proceso | Se realiza una conexión remota al sistema gestor de base de datos del servidor para ejecutarla y obtener los datos por medio de un archivo Json. Con el resultado de la consulta se crear la tabla dinámica y se divide en diferentes páginas. |
| Situación anormal | Que la conexión con el servidor se pierda, el tiempo de espera para procesar dicha solicitud acabe y por último que la sentencia no se pueda procesar por errores de sintaxis. |
| Precondición | Mantener el dispositivo conectado con la nube en el instante que se procesa la sentencia y se adquiere el archivo. |
| Post-Condición | Continuar con la conexión al servidor remoto en caso que la sentencia rebase el número de filas observables en pantalla o se requiera actualizar la tabla. |
| Efecto colateral | El tiempo de espera para la creación de la tabla dependerá del número de columnas y filas que la sentencia genere. |
| Prioridad | Alta |
| Rol | Administrador |

### Requisito Funcional Número Siete

|  |  |
| --- | --- |
| Id requerimiento | 7 |
| Descripción | Graficación de la sentencia. |
| Entradas | Datos de acceso obligatorios: id del servidor alojado, base de datos seleccionada, tipo de sentencia y sentencia. |
| Salidas | Gráfica de los campos en base a un campo base. |
| Proceso | Se realiza una conexión remota al sistema gestor de base de datos del servidor para obtener las columnas, después desplegar a pantalla un listado de las columnas, se selecciona el campo base y posteriormente se realiza una nueva conexión con el servidor, generando un archivo Json con los datos de las columnas a graficar. Al recibir el archivo la aplicación grafica los datos obtenidos. |
| Situación anormal | Que el tiempo de espera para visualizar la gráfica a pantalla sea demasiado o que la cantidad de datos a graficar sea exorbitante. |
| Precondición | Mantener el dispositivo conectado con la nube mientras se procesa y se obtienen el archivo Json. |
| Post-Condición | El dispositivo móvil debe de contar con una memoria SD en caso de guardar la imagen de la gráfica. |
| Efecto colateral | El tiempo de espera para la creación de la gráfica dependerá del número filas que la sentencia genere y del número de campos en relación al campo base. Se podrá almacenar una imagen de la gráfica en la memoria SD. |
| Prioridad | Alta |
| Rol | Administrador |

## Requisitos No Funcionales

### Requisitos De Rendimiento

Al ser una aplicación móvil el número de usuarios se limita a uno por dispositivo, por otra parte el tiempo de respuesta está completamente ligado a la velocidad del ancho de banda del usuario, aun así se espera que el tiempo máximo de respuesta sea de 3 a 5 segundos por petición. Mientras que el tipo para general la tabla y la gráfica dependerá del número de datos.

### Seguridad Y Validación De Información

Se cifraran los datos de la conexión con el servidor, además de que el sistema podrá realizar sentencias SQL de tipo SELECT, UPDATE, DELETE, ADD, solo si el usuario cuenta con tales permisos dentro de la base remota. El sistema debe validar toda la información que el usuario ingrese, como es agregar datos del gestor, editar datos del gestor y la sintaxis de la sentencia.

### Disponibilidad E Instalación

Dado que la aplicación de un inicio se está proponiendo para un proceso de plasma no térmico por parte de la Universidad Autónoma del Estado de México, la App solo estará disponible a los miembros del equipo de trabajo. Por lo cual no estará de momento disponible al público en Google Play. La instalación se realizara directamente en dispositivos que cumplan con los requisitos de hardware y software.

# Capítulo IV - Medición Y Modelado De La Aplicación

## Diseño Del Sistema

Un diseño de software es la descripción de la estructura del software que se va a implementar, el proceso de diseñar puede implicar el desarrollo de varios modelos del sistema con distintos niveles de abstracción.

Para el proceso de modelación se usó UML, el Lenguaje de Modelado Unificado (Unified Modeling Language) es sucesor de la oleada de métodos y análisis y diseño orientado a objetos (OOA&D). El lenguaje de modelado es la notación gráfica de que se valen los métodos para expresar los diseños (Fowler Martin & Scott, 1999).

### Casos De Uso

Los diagramas de casos de uso son una técnica para capturar los requerimientos funcionales de un sistema. Se enfocan en la interacción entre los usuarios del sistema y el mismo sistema. En la figura 4.1 se muestran el diseño del diagrama de casos de uso de la aplicación SQL Graphic, dicho diagrama ejemplifica los procesos a los cuales el usuario tiene acceso directo como son los módulos de agregar, editar o eliminar datos del sistema gestor de base datos entre otros más.

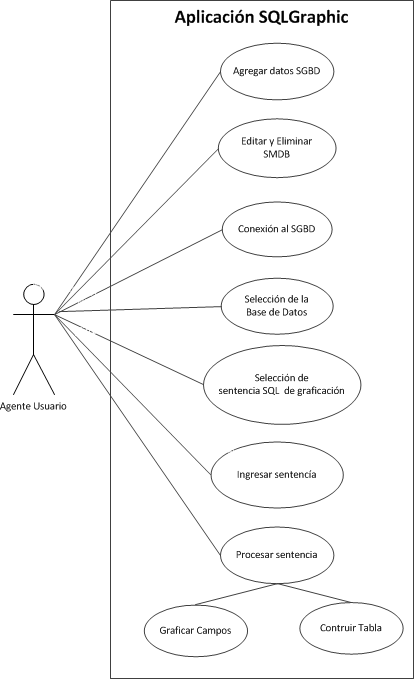


Figura 4. - Diagrama de casos de uso de la Aplicación SQLGraphic.

En la figura 4.2 se modelo el subdiagrama de casos de uso de la aplicación, ajustando las acciones correspondientes del dispositivo; al interactuar el usuario de manera directa o indirectamente con los procesos externos e internos del sistema de la aplicación móvil. Algunos procesos extienden información a otros procesos o usan algunos métodos derivados de procesos internos.

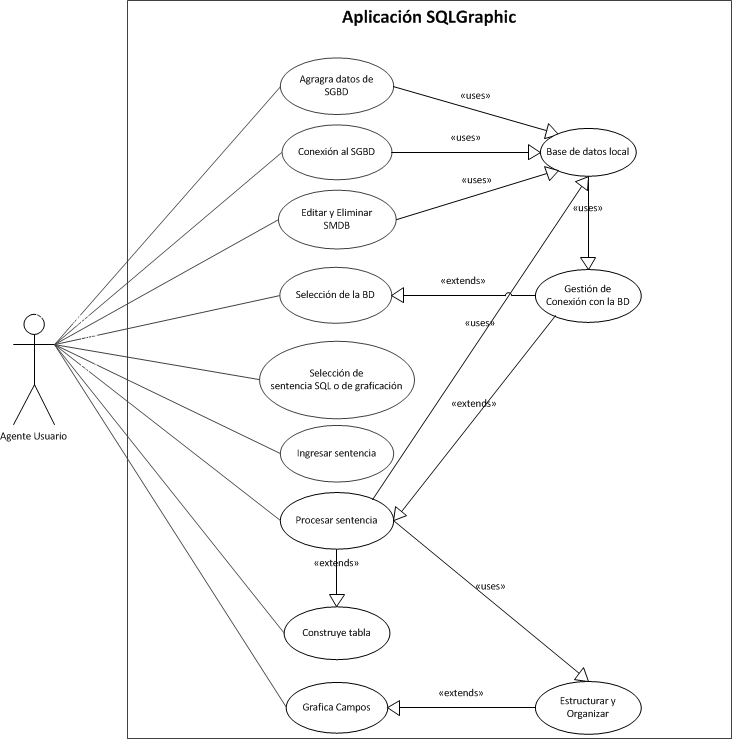


Figura 4. - Subdiagrama de casos de uso de Aplicación SQLGraphic.

### Diagrama De Secuencia

Los diagramas de secuencia ilustran una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Cada uno de los escenarios de casos de uso podría crear un de estos diagramas.

**Agregar Datos Y Autenticación Al Servidor**

Inicialmente se agregan los datos del servidor en la base local de la aplicación, los cuales son nombre local, host, puerto, nombre de usuario y contraseña, para lograr autentificarse en el servidor el agente móvil controlador será el intermediario entre el SGBD y la GUI, como se muestra en la figura 4.3, para dar de alta y consultar los datos del servidor en la base local del dispositivo móvil.

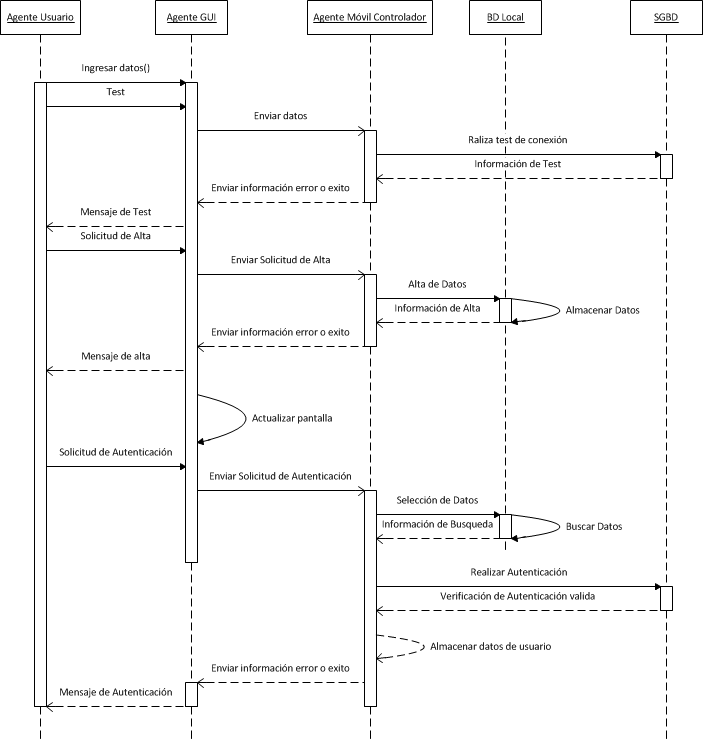


Figura 4. - Diagrama de secuencia de autenticación en el servidor.

**Acceso a la base de datos**

Una vez autentificado el usuario en el sistema gestor de base de datos el agente móvil controlador se encargara de guardar sus credenciales, para después realizar la consulta de las bases de datos que integran al sistema administrador o a las que tiene acceso el usuario. Como resultado de dicha consulta se inicia la actividad generar de la interfaz de usuario, en la cual se podrá seleccionar la base de datos si no se definió previamente al dar de alta los datos del servidor en la base de datos local. Por último se almacenar la información de la base de datos en la actividad principal como se modela en la figura 4.4.



Figura 4. - Diagrama de elección de la base de datos.

**Procesar secuencia de tipo tabla**

El agente móvil ordena las tablas en orden alfabético y crea en un campo de selección un listado de las tablas que conforman la base de datos, el usuario hasta este punto puede seleccionar la tabla a utilizar o ingresar una sentencia de tipo tabla como se hace en otros programas de manejo de servidores remotos. La figura 4.5 muestra el flujo de acciones para que se procesar este tipo de sentencias.

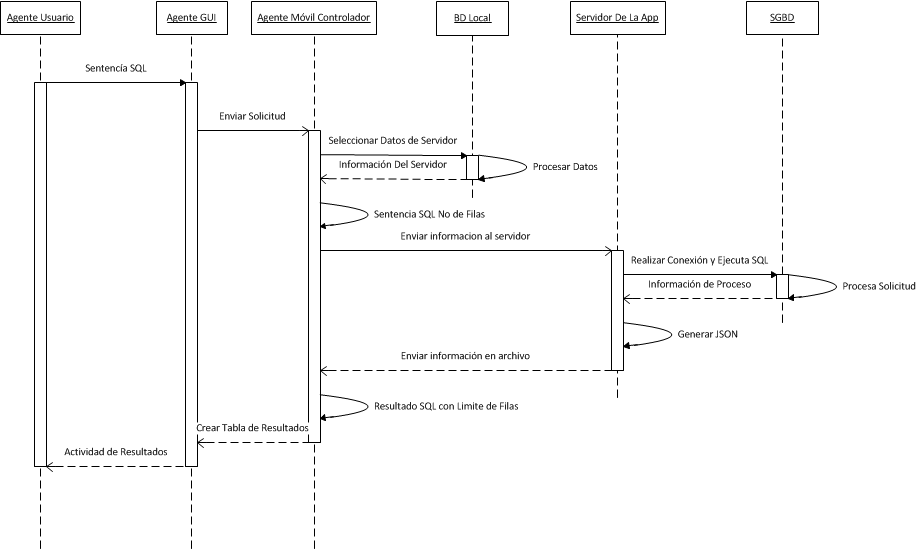


Figura 4. - Diagrama de secuencia de sentencia SQL.

**Sentencia de Graficación**

La sentencia permite graficar diversos campos de la misma tabla en base a un campo base, el cual puede estar alojado en otra tabla de la misma base de datos. Finalmente se realiza se procesa los datos usando un servidor externo para aumentar la velocidad de procesamiento de la información y graficación, como se muestra en la figura 4.6.

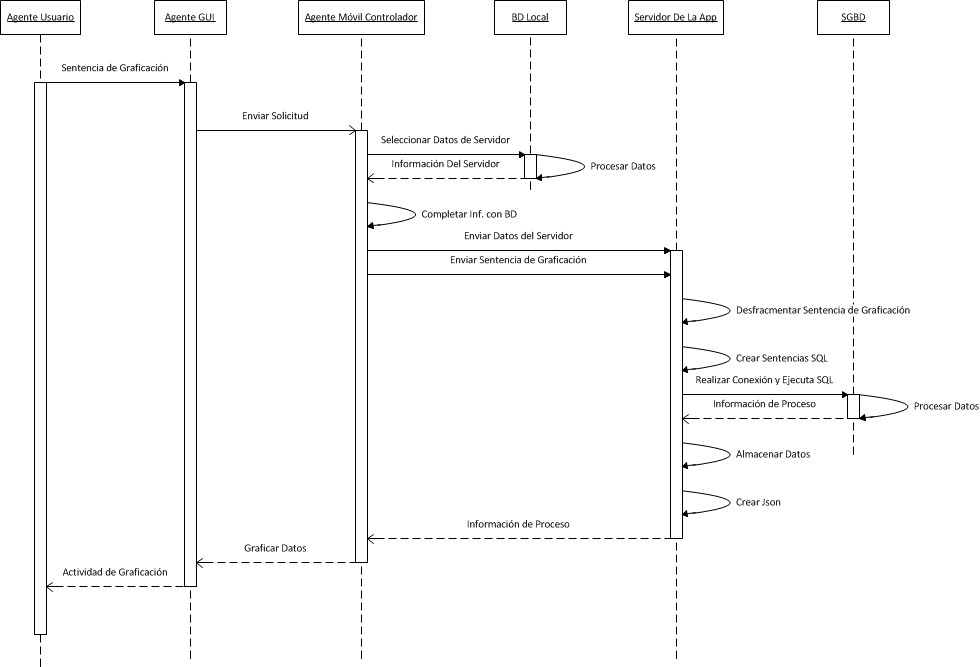


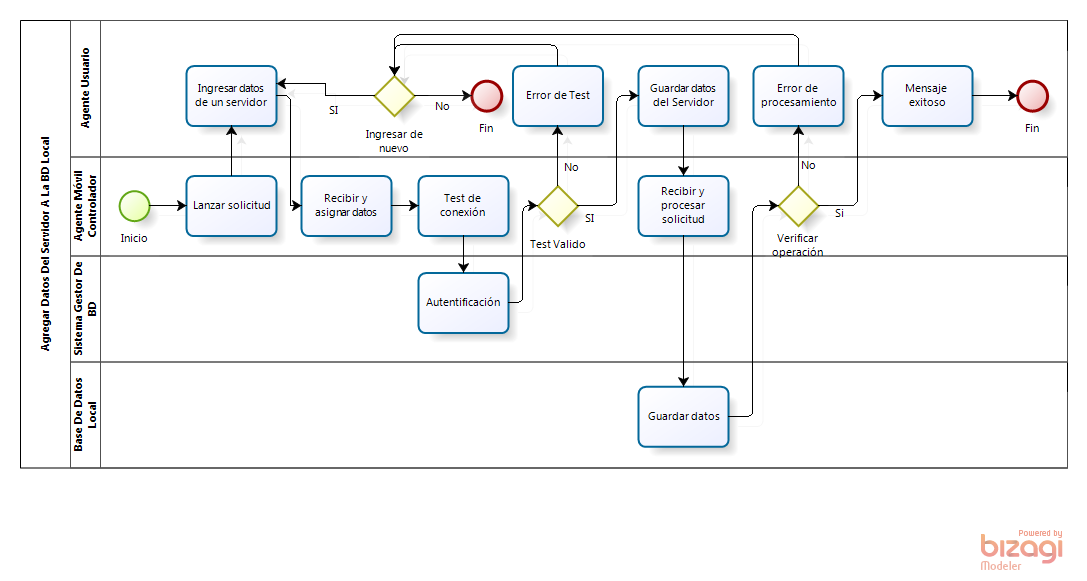
Figura 4. - Diagrama de sentencia de graficación.

### Diagrama De Actividad

Los diagramas de actividad se definen como una técnica de modelado que describe un proceso lógico, un proceso de negocios o un flujo de trabajo paso a paso. Además de ofrecer la posibilidad de modelar procesos que se realizan de manera paralela (Fowler, 2004).

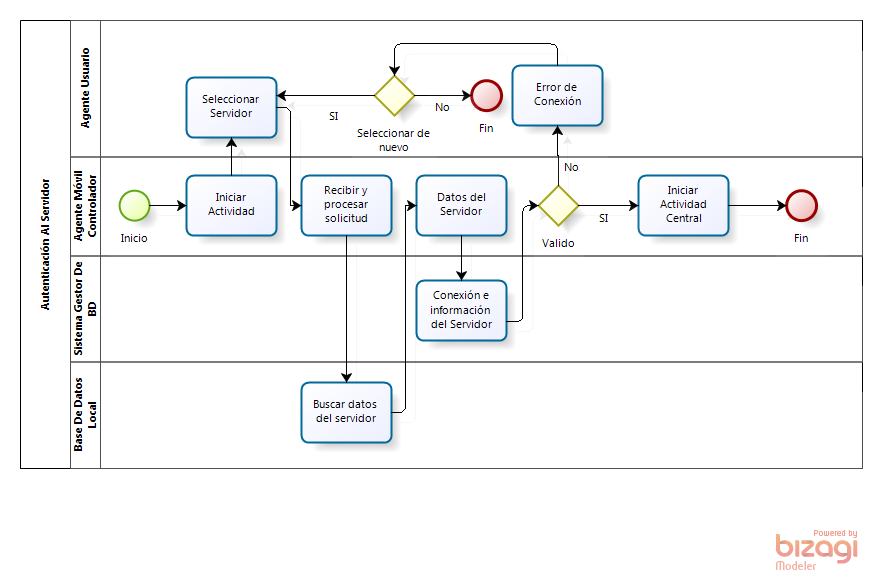
**Agregar Datos Del Servidor A La BD Local**

La figura 4.7 representa la secuencia de actividades a realizar entre el agente móvil y el agente usuario para dar de alta los datos del servidor en la base local del dispositivo móvil, además de permitir realizar la función de testeo para verificar la conectividad con el servidor.

Figura 4. - Diagrama de actividad para agregar datos del servidor a la BD local.

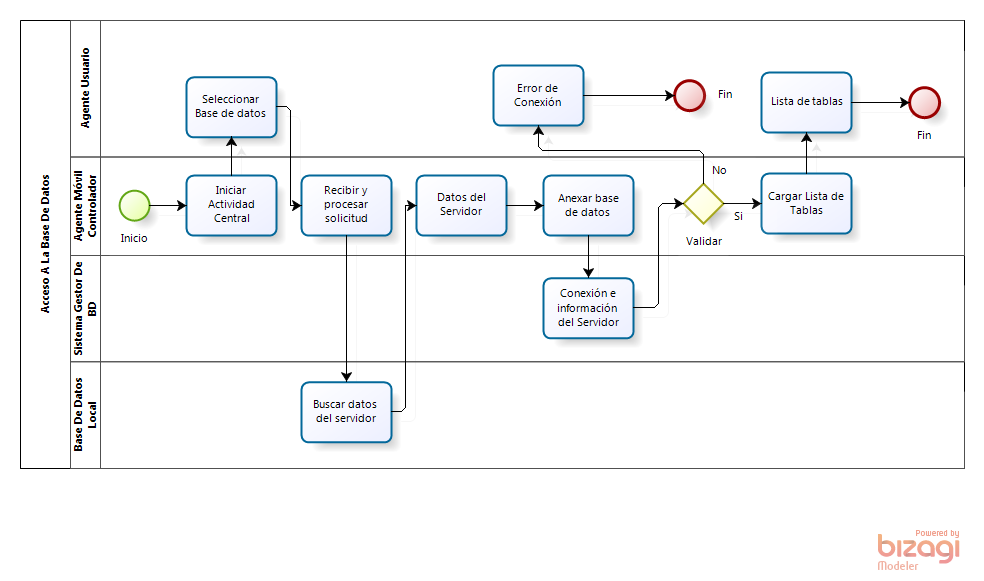
**Autenticación Al Servidor**

Para realizar la autenticación y tener acceso a la información del servidor se requiere tener alojados los datos de conexión en el depósito de información local, como se muestra en la sucesiva figura.

Figura 4. - Diagrama de actividad de autenticación al servidor.

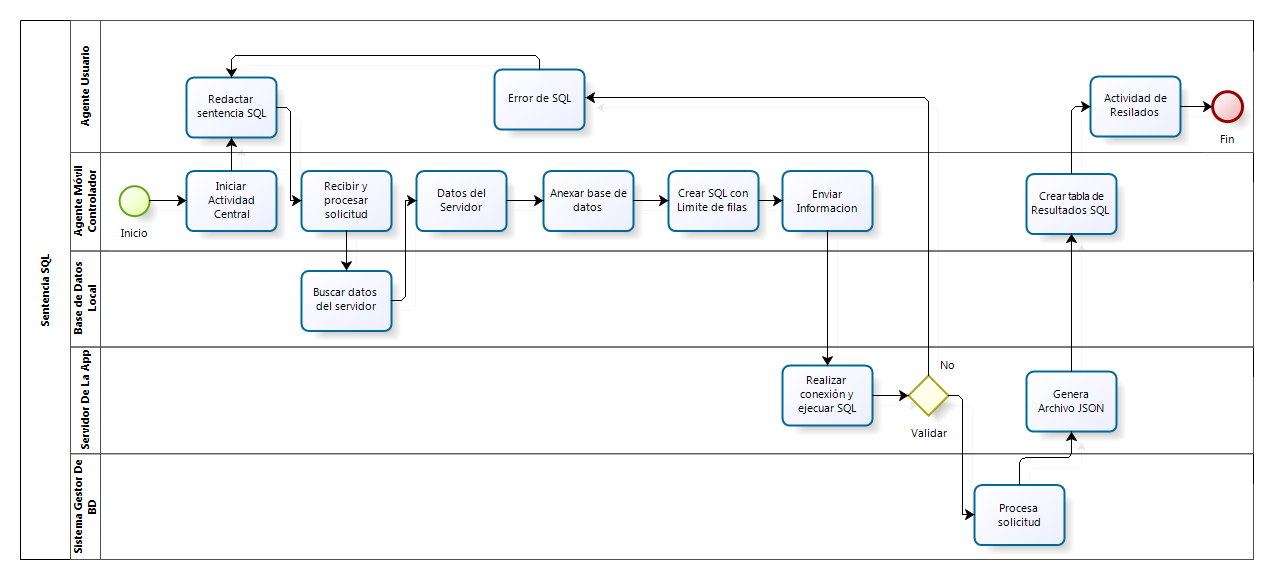
**Diagrama De Actividad De Selección De La Base De Datos**

Al iniciar la actividad central el usuario debe seleccionar la base de datos a la cual se conectara de lo contrario no podrá ejecutar ninguna sentencia de tipo tabla o de graficación. De manera paralela se solicita información sobre las tablas que forman la base de datos, expuesto en la figura 4.9.

Figura 4. - Diagrama de actividad de selección de BD.

**Diagrama De Actividad De Sentencia De Tabla**

Las sentencias de tipo tabla serán ejecutadas directamente en el servidor remoto, antes que nada se habrá que seleccionar este tipo de sentencia y haber seleccionado la base de datos donde se ejecutara, los resultados se presentara en una tabla dinámica que depende de la función ejecutada, se muestra la secuencia de actividades en la siguiente figura.

Figura 4. - Diagrama de actividad de sentencia de tabla.

**Diagrama De Actividad De Sentencia De Graficación**

La sentencia de graficación se ejecutara en el servidor de la aplicación por cuestión de rendimiento, para ello la App debe enviar la información del servidor y la sentencia de graficación, la cual se dividirá en sentencias SQL, los resultados arrojados por esta sentencia se normalizaran para luego graficarse como se observa a continuación en el figura 4.11.



Figura 4. -Diagrama de actividad de sentencia de graficación.

### Diagrama De Clases

Los diagramas de clases muestran la estructura estática del sistema y no representa ningún proceso, se enfoca en describir las clases, atributos, métodos y la relación entre las clases. Este tipo de diagramas utiliza el análisis orientado a objetos para definir las clases asociadas al sistema (Kenneth, 2005).

La figura 4.12 muestra la estructura de la aplicación mostrando sus clases, atributos, métodos y relaciones entre clases de la aplicación SQLGraphic en dispositivos android.

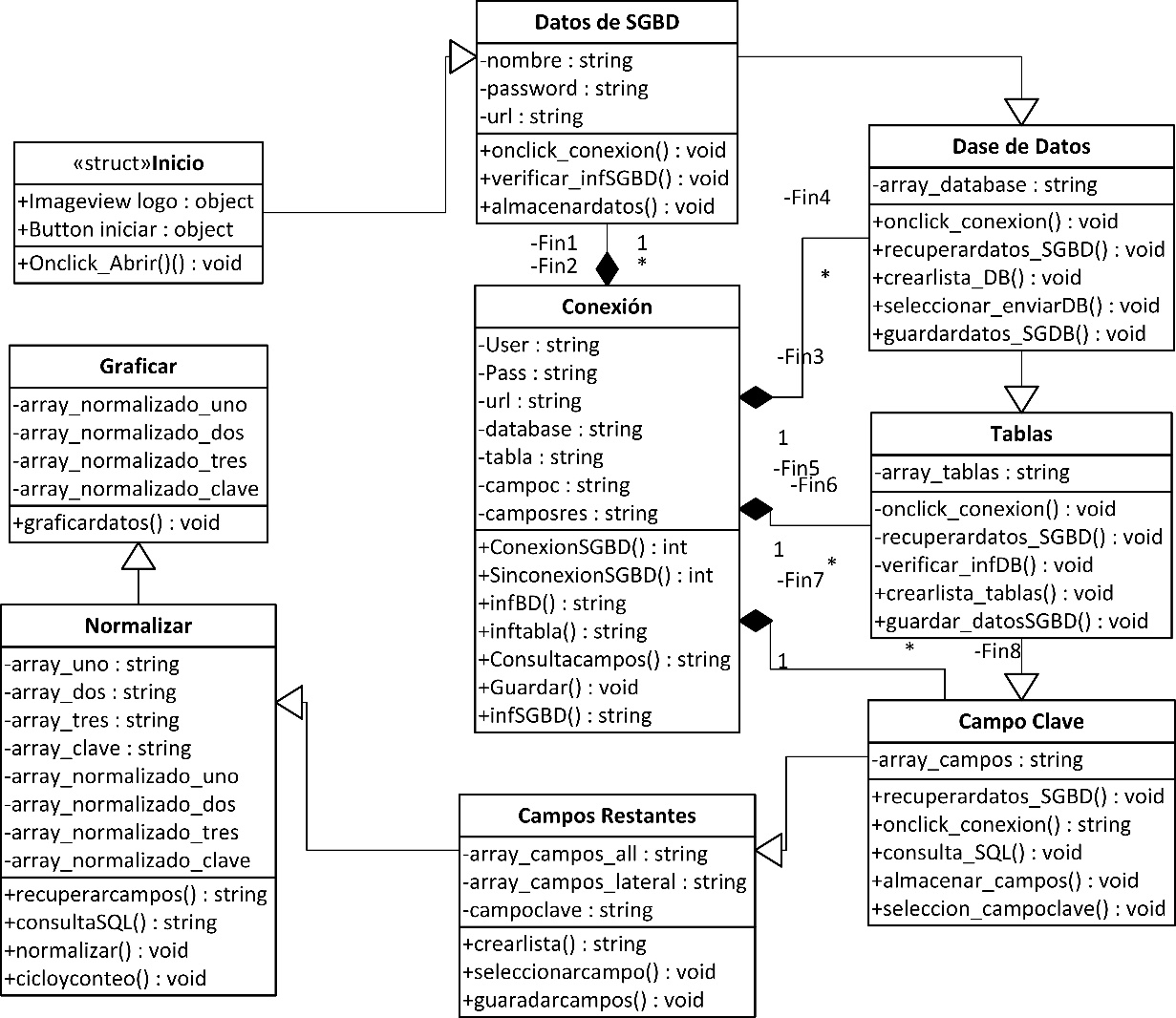


Figura 4. - Diagrama de clases.

## Modelo Constructivo De Costos II

El modelo constructivo de costos (COCOMO) surge en el año de 1981 por Barry Boehm, años posteriores se readapto para el desarrollo de software de ADA (Boehm y Royce, 1989), y finalmente se actualiza en el año 2000 denominado como COCOMO II. Para más información acerca de fórmulas y tablas de valores consultar Anexo 7.

### Estimación De La Aplicación

La estimación del proyecto SQLGraphic entra en el modelo anticipado o temprano, este modelo calcula el esfuerzo usando la ecuación 1:

Ecuación - Esfuerzo nominal ajustado

* es el esfuerzo nominal ajustado por 7 factores.
* corresponde a los factores de costo que multiplican el esfuerzo llamados Multiplicadores de Esfuerzo (*Effort Multipliers*) clasificados en:
* Del producto
* RCPX: confiabilidad y complejidad del producto.
* RUSE: reusabilidad requerida
* De la plataforma:
* PDIF: dificultad de la plataforma.
* Del personal
* PERS: aptitud del personal.
* PREX: experiencia del personal.
* Del proyecto
* FCIL: facilidades.
* SCED: cronograma de desarrollo requerido.
* es el esfuerzo nominal sin ajustar y es calculado con la ecuación 2.

Ecuación - Esfuerzo nominal sin Ajustar

* A es una constante de variación en base al tamaño y es igual a 2.94.
* KSLOC son miles de líneas de código fuente.
* B es el factor exponencial de escala relacionado con la productividad.

Para calcular inicialmente PM nominal se requiere de calcular el número de Kilo-líneas de código fuente utilizando puntos de función o puntos objetos, un experimento diseñado por Kauffman y Kumar en 1993, permitió determinar que los puntos objeto y los puntos de función producen resultados igualmente precisos.

Los puntos de función se obtienen por medio de la ecuación 3.

Ecuación - Puntos de Función (FP)

* son Puntos de Función no Ajustados.
* es el factor de Complejidad Técnica.

Para calcular los puntos de función sin ajustar se utilizan los siguientes ítems:

* Entradas Externas (EI): entradas de datos del usuario o de control que se ingresan desde el exterior del sistema.
* Salidas Externas (EO): salida de datos de usuario o de control generadas por el sistema.
* Consultas Externas (EQ): combinación única de entrada y salida donde una entrada genera una salida inmediata.
* Archivos Lógicos Internos (ILF):incluye cada archivo lógico
* Archivos Externos de Interface (EIF): archivos transferidos o compartidos entre sistemas de software.

Después de identificar los ítems se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto, para asignar un peso según el grado de complejidad, para calcular los se emplea la ecuación 4:

Ecuación - Puntos de Función no Ajustados (UFP)

En la tabla 4.1 se muestran cómo se determinan los niveles de complejidad de cada tipo de ítem de acuerdo a la función del número, tipo de elemento y archivos involucrados.

Tabla 4. - Niveles de complejidad de cada tipo de ítem.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Para ILF y EIF** | | | | **Para OE y QE** | | | | **Para IE** | | | |
| **Elementos de registro** | **Elementos de datos** | | | **Tipos de archivos** | **Elementos de datos** | | | **Tipos de archivos** | **Elementos de datos** | | |
| **1 a 19** | **20 - 50** | **51+** | **1 a 5** | **6 a 19** | **20+** | **1 a 4** | **5 a 15** | **16+** |
| **1** | Bajo | Bajo | Prom. | **0 ó 1** | Bajo | Bajo | Prom. | **0 ó 1** | Bajo | Bajo | Prom. |
| **2 a 5** | Bajo | Prom. | Alto | **2 a 3** | Bajo | Prom. | Alto | **2 a 3** | Bajo | Prom. | Alto |
| **6+** | Prom. | Alto | Alto | **4+** | Prom. | Alto | Alto | **3+** | Prom. | Alto | Alto |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

Mientras que en la tabla 4.2 se muestra las ponderaciones asociadas a cada tipo de ítem. Estas ponderaciones han sido derivadas y validadas empíricamente mediante observación de una gran variedad de proyectos.

Tabla 4. - Peso del factor de complejidad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de Función** | **Bajo** | **Promedio** | **Alto** | **Total** |
| **EI** | 3 | 4 | 6 | - |
| **EO** | 4 | 5 | 7 | - |
| **EQ** | 3 | 4 | 6 | - |
| **ILF** | 3 | 10 | 15 | - |
| **EIF** | 7 | 7 | 10 | - |
| **Total** | - | - | - | - |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

Para calcular el Factor de Complejidad Técnica (TCF) es considerada la siguiente ecuación:

Ecuación - Factor de Complejidad Técnica (TCF)

Donde corresponde al peso asignado por los siguientes factores

* Mecanismo de recuperación y back-up confiables.
* Comunicación de datos.
* Funciones de procesamiento distribuido.
* Performance.
* Configuración usada rigurosamente.
* Entradas de datos on-line.
* Factibilidad operativa.
* Actualización de archivos.
* Interfaces complejas.
* Procesamiento interno complejo.
* Reusabilidad.
* Fácil instalación.
* Soporte de múltiples instalaciones.
* Facilidad de cambios y amigabilidad.

Los pesos se consideran en una escala del 0 a 5, 0 – Sin influencia, 1 – Incidental, 2 – Moderado, 3 – Promedio, 4 – Significativo y 5- Esencial. Para calcular los puntos de función de la aplicación móvil SQLGraphic se asignan valores a cada uno de los ítems para obtener los puntos de función no ajustados en base al SRS del capítulo III, los cuales se muestran en la tabla siguiente.

Tabla 4. - Factor de complejidad de la aplicación SQLGraphic.

|  |  |
| --- | --- |
| Tipo de función | Valor de asignación |
| Entradas Externas (EI) | 2 |
| Salidas Externas (EO) | 2 |
| Consultas Externas (EQ) | 4 |
| Archivos Lógicos Internos (ILF) | 3 |
| Archivos Externos de Interface (EIF) | 4 |

Luego se determinan los niveles de complejidad para cada tipo de ítem que conforman el factor de complejidad de la aplicación, en base a los requerimientos del capítulo III. La asignación del factor se aprecia en la tabla 4.4.

Tabla 4. - Determinación del grado de complejidad de cada ítem.

|  |  |
| --- | --- |
| Tipo de función | Peso de asignación |
| Entradas Externas (EI) | Bajo |
| Salidas Externas (EO) | Bajo |
| Consultas Externas (EQ) | Promedio |
| Archivos Lógicos Internos (ILF) | Bajo |
| Archivos Externos de Interface (EIF) | Promedio |

Al tener el grado de complejidad asignado a cada uno de los ítems se prosigue sustituirlos por los valore de la tabla 4.2, para después ser multiplicados por el número de ítems que se encuentran representados en la tabla 4.3, como se puede apreciar a continuación en la tabla.

Tabla 4. - Tabla de puntos de función no ajustados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de Función** | **Valor de Asignación** | **Nivel de complejidad** | **Multiplicación** | **Total** |
| **EI** | 2 | 3 | (2)(3) | 6 |
| **EO** | 2 | 4 | (2)(4) | 8 |
| **EQ** | 4 | 4 | (4)(4) | 16 |
| **ILF** | 3 | 3 | (3)(3) | 9 |
| **EIF** | 4 | 7 | (4)(7) | 28 |
| **Total** |  |  |  | 67 |

En base a la tabla 4.5 se obtiene el valor del total de puntos de función no ajustados, que es de 67 UFP. Ahora se calcula el Factor de Complejidad Técnica (TCF), para lo cual asignamos el grado a cada uno de los catorce factores de acuerdo al grado de influencia en el sistema, como se muestra en la siguiente tabla.

Tabla 4. - Factores de Ajuste de TCF.

|  |  |  |
| --- | --- | --- |
| Factores de ajuste | Nivel | Valor |
| Mecanismo de recuperación y back-up confiables. | Moderado | 2 |
| Comunicación de datos. | Esencial | 5 |
| Funciones de procesamiento distribuido. | Promedio | 3 |
| Performance. | Esencial | 5 |
| Configuración usada rigurosamente. | Incidental | 1 |
| Entradas de datos on-line. | Esencial | 5 |
| Factibilidad operativa. | Esencial | 5 |
| Actualización de archivos. | Moderado | 2 |
| Interfaces complejas. | Incidental | 1 |
| Procesamiento interno complejo. | Promedio | 5 |
| Reusabilidad. | Significativo | 4 |
| Fácil instalación. | Esencial | 5 |
| Soporte de múltiples instalaciones. | Sin influencia | 0 |
| Facilidad de cambios y amigabilidad. | Significativo | 4 |
| Total |  | 47 |

Se procede a emplear la ecuación 5 para calcular el factor de complejidad técnica, sustituyendo la sumatoria de todos los factores de acuerdo al nivel de prioridad que se les asigno en la tabla 4.6.

Ecuación - Factor de Complejidad Técnica (TCF) del proyecto SQLGraphic.

Después de calcular el factor de complejidad técnica que es de 1.12 TCF y los puntos de función sin ajustar, se sustituyen ambos valores en la ecuación 3 para obtener los puntos de función ajustados que son de 141.04 como se aprecia en la siguiente ecuación.

Ecuación - Puntos de Función (FP) del sistema SQLGraphic.

Para determinar el esfuerzo nominal en COCOMO II los puntos de función tienen que ser convertidos a líneas de código fuente, se toma en considerando el lenguaje a utilizar para la implementación de la aplicación (assemble, lenguaje de alto nivel, lenguaje de cuarta generación, etc.) esto es utilizado en modelado de diseño temprano y post arquitectura. Para calcular las miles de líneas de código fuente se utiliza la ecuación 8.

Ecuación - Miles de líneas de código fuente.

Donde FP son los puntos de función y SLOC son las líneas de código fuente, se utiliza la siguiente tabla para definir SLOC de acuerdo al lenguaje que se usa para programar la aplicación móvil.

Tabla 4. - Conversión de FP a SLOC con factor de apalancamiento.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lenguaje | QSM SLOC/FP Data | | | |
| **Promedio** | **Medio** | **Bajo** | **Alto** |
| ABAP (SAP) | 28 | 18 | 16 | 60 |
| ASP | 51 | 54 | 15 | 69 |
| Assembler | 119 | 98 | 25 | 320 |
| Brio | 14 | 14 | 13 | 16 |
| C | 97 | 99 | 39 | 333 |
| C++ | 50 | 53 | 25 | 80 |
| C# | 54 | 59 | 29 | 70 |
| COBOL | 61 | 55 | 23 | 297 |
| Cognos Impromptu Scripts + | 47 | 42 | 30 | 100 |
| Cross System Products (CSP) + | 20 | 18 | 10 | 38 |
| Cool:Gen/IEF | 32 | 24 | 10 | 82 |
| Datastage | 71 | 65 | 31 | 157 |
| Excel | 209 | 191 | 131 | 315 |
| Focus | 43 | 45 | 45 | 45 |
| FoxPro | 36 | 35 | 34 | 38 |
| HTML | 34 | 40 | 14 | 48 |
| J2EE | 46 | 49 | 15 | 67 |
| Java | 53 | 53 | 14 | 134 |
| JavaScript | 47 | 53 | 31 | 63 |
| JCL | 62 | 48 | 25 | 221 |
| LINC II | 29 | 30 | 22 | 38 |
| Lotus Notes | 23 | 21 | 19 | 40 |
| Natural | 40 | 34 | 34 | 53 |
| .NET | 57 | 60 | 53 | 60 |
| Oracle | 37 | 40 | 17 | 60 |
| Perl | 24 | 15 | 15 | 60 |
| PL/1 | 64 | 80 | 16 | 80 |
| PL/SQL | 37 | 35 | 13 | 60 |
| Powerbuilder | 26 | 28 | 7 | 40 |
| REXX | 77 | 80 | 50 | 80 |
| Sabretalk | 70 | 66 | 45 | 109 |
| SAS | 38 | 37 | 22 | 55 |
| Siebel | 59 | 60 | 51 | 60 |
| SLOGAN | 75 | 75 | 74 | 75 |
| SQL | 21 | 21 | 13 | 37 |
| VB.NET | 52 | 60 | 26 | 60 |
| Visual Basic | 42 | 44 | 20 | 60 |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

El factor de apalancamiento es simplemente el número promedio de líneas de código fuente (SLOC) por punto de función en el proyecto terminado. Originalmente, fue diseñado para ser utilizado como un punto de referencia común para la comparación de diferentes métricas de tamaño mediante la asignación a la unidad de tamaño más pequeño común a todos los proyectos de software. QSM[[7]](#footnote-7) recomienda que las organizaciones recogen ambos aspectos de código y puntos de conteo función final de los proyectos de software completos y utilizan estos datos para las estimaciones nuevas. Para el proyecto SQLGraphic se utilizara java como lenguaje de programación y un factor de apalancamiento promedio, se sustituye ambos datos en la ecuación 8 como se observa en la ecuación 9.

Ecuación - Total de miles de líneas de código fuente (KSLOC) para aplicación SQLGraphic

El factor exponencial de escala se utiliza en los modelos de estimación de costos para considerar las economías y deseconomías de escala. En particular en COCOMO II se capturan estos efectos en el componente B y el cual se calcula con la sucesiva ecuación.

Ecuación – Factor Exponencial De Escala

* Si , el proyecto exhibe economía de escala, la productividad del proceso de desarrollo de software incrementa a medida que aumenta el tamaño del proyecto.
* Si , la economía y deseconomías están en escala de equilibrio. Este modelo se usa en estimaciones de proyectos pequeños.
* Si , el proyecto muestra deseconomía de escala, la productividad del proceso de desarrollo de software disminuye a medida que el tamaño aumenta.

El factor exponencial de escala está basado en factores básicos que influye exponencialmente en la productividad y esfuerzo del proyecto, estos factores son evaluados con valores de rango que van desde muy bajo hasta extra alto como se ve en la siguiente tabla.

Tabla 4. - Factores exponenciales de escala.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factores de Escala | MUY BAJO | BAJO | NORMAL | ALTO | MUY ALTO | EXTRA ALTO |
| Precedencia PREC | Completamente sin precedentes | Ampliamente sin precedentes | Algún precedente | Generalmente Familiar | Ampliamente Familiar | Completamente Familiar |
| Flexibilidad En El Desarrollo FLEX | Rigurosa | Relajación ocasional | Alguna Relajación | Conformidad General | Alguna Conformidad | Metas Generales |
| Arquitectura/Resolución De Riesgo RESL | Poca (20%) | Alguna (40%) | Siempre (60%) | Generalmente (75%) | Principalmente (90%) | Completo (100%) |
| Cohesión De Equipo TEAM | Interacciones Difíciles | Interacciones con alguna dificultad | Interacciones Básicamente Cooperativas | Ampliamente cooperativas | Altamente Cooperativo | Interacciones sin Fisuras |
| Madurez Del Proceso PMAT | Desarrollado más adelante | | | | | |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

En la anterior tabla se observa los pesos de cada factor según el nivel de influencia, establecidos por COCOMO II, mientras que en la tabla 4.9 se muestran los datos numéricos de acuerdo al factor de escala.

Tabla 4. - Factor de escala de COCOMO II.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factores de Escala | MUY BAJO | BAJO | NORMAL | ALTO | MUY ALTO | EXTRA ALTO |
| Precedencia PREC | 6.20 | 4.96 | 3.72 | 2.48 | 1.24 | 0.00 |
| Flexibilidad En El Desarrollo FLEX | 5.07 | 4.05 | 3.04 | 2.03 | 1.01 | 0.00 |
| Arquitectura/Resolución De Riesgo RESL | 7.07 | 5.56 | 4.24 | 2.83 | 1.41 | 0.00 |
| Cohesión De Equipo TEAM | 5.48 | 4.38 | 3.29 | 2.19 | 1.10 | 0.00 |
| Madurez Del Proceso PMAT | 7.80 | 6.24 | 4.68 | 3.12 | 1.56 | 0.00 |

Para el sistema SQLGraphic se asignaron los siguientes valores de acuerdo en la tabla 4.8 y 4.9, el factor PMAT se fija según el nivel CMM[[8]](#footnote-8) de la tabla 4 del anexo 7, siguiente tabla es resultado asignación de valores, para finalmente ser sustituidos por los valores de escala de COCOMO II.

Tabla 4. - Factores exponenciales de escala del SQLGraphic.

|  |  |  |
| --- | --- | --- |
| **Factores de Escala** | **NIVEL ASIGNADO** | **COCOMO II** |
| Precedencia PREC | BAJO | 4.96 |
| Flexibilidad En El Desarrollo FLEX | NORMAL | 3.04 |
| Arquitectura/Resolución De Riesgo RESL | MUY BAJO | 7.07 |
| Cohesión De Equipo TEAM | MUY ALTO | 1.10 |
| Madurez Del Proceso PMAT | BAJO | 4.68 |
| TOTAL |  | 20.85 |

Se sustituyen el valor obtenido del factor exponencial de escala en la ecuación 10 como se observa en la siguiente ecuación.

Ecuación – Factor Exponencial De Escala De SQLGraphic.

Se reemplaza el factor exponencial de escala (B) y las miles de líneas de código fuente (KSLOC) en la ecuación 2 para obtener el calcular el esfuerzo nominal del sistema SQLGraphic en la ecuación 13.

Ecuación – Esfuerzo Nominal sin ajustar de SQLGraphic

Para calcular el tiempo de desarrollo se utiliza la ecuación 3 y es expresado en meses. Tomando en consideración los factores de escala exponencial, el esfuerzo nominal y excluyendo el factor de esfuerzo SCED, genera la ecuación 14.

Ecuación – Estimación Temporal

* es la estimación temporal de desarrollo del proyecto.
* es el esfuerzo nominal.
* es el factor de escala exponencial.

Se sustituye los datos del esfuerzo nominal y el factor de escala exponencial en la ecuación 14 de estimación temporal como se observa en la siguiente ecuación.

Ecuación - Estimación Temporal De SQLGraphic

Para calcular el esfuerzo nominal ajustado en un parámetro de costes combinados del modelo de diseño inicial, los valores numéricos de los parámetros de coste del modelo post-arquitectura son sumados y el resultado total es asignado a un ratio de escala del modelo de diseño inicial de la tabla siguiente.

Tabla 4. - Multiplicadores De Esfuerzo Diseño Inicial Y Post-Arquitectura.

|  |  |
| --- | --- |
| Parámetro de Coste | Combinación Equivalente |
| Diseño Inicial | Post-Arquitectura |
| RCPX | RELY, DATA, CPLX, DOCU |
| RUSE | RUSE |
| PDIF | TIME, STOR, PVOL |
| PERS | ACAP, PCAP, PCON |
| PREX | AEXP, PEXP, LTEX |
| FCIL | TOOL, SITE |
| SCED | SCED |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

# Capítulo V - Diseño De Los Componentes De La Aplicación

## Diseño De Base De Datos

El diseño de base de datos se refiere a las actividades que se concentrar en el diseño de la estructura de la base de datos que se usara para guardar la información del usuario final. Las relaciones entre tablas deben ser cuidadosamente consideradas e implementadas para que la vista integral de los datos se recreada como información al usuario final.

### Modelo De Entidad Relación

Los modelos o diagramas de entidad relación es una herramienta para el diseño de base de datos, los elementos claves de este tipo de diagramas son las entidades, atributos, identificadores y relaciones, en la figura siguiente se observa el diagrama de entidad relación de notación tipo Chen del sistema SQLGraphic.



Figura 5. - Diagrama De Entidad Relación Notación Tipo Chen.

En la figura 5.2 se ilustra el modelo entidad relación de notación tipo “pata de gallo” del sistema SQLGraphic, el nombre se deriva del símbolo de tres puntas que se utiliza para representar la relación de uno a muchos. Existe una relación de uno a muchos (1: M) entre la tabla DataServer y Sentence, donde un servidor puede realizar ejecutar muchas sentencias de tipo SQL o de graficación y donde cada sentencia solo es ejecutada por un servidor. Además la tabla Graphic y Sentence comparten una relación de uno a uno (1: 1), donde una gráfica solo es creada por una sentencie y de manera inversa.

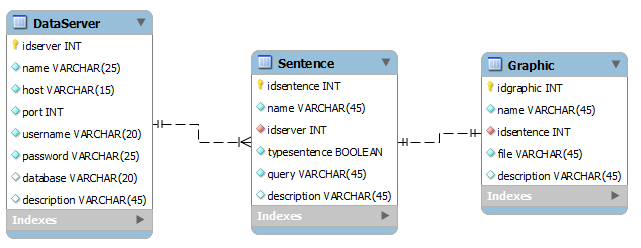


Figura 5. - Diagrama De Entidad Relación Notación Tipo “Pata de Gallo”.

### Diccionario De Datos

El diccionario de datos es una descripción detallada de las tablas que se encuentran dentro de las base de datos. Contiene al menos todos los nombres de los atributos y características para cada tabla del sistema. El diccionario de datos del sistema SQLGraphic se presenta en las tablas siguientes.

Tabla 5. - Diccionario de datos de SQLGraphic.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la Tabla | | DataServer | | | | |
| Nombre de los Campos | | **Tipo de Dato** | **Contenido** | **PK FK** | **Formato** | **Requerida** |
| 1 | idserver | INT | Id del servidor. | PK | 9999 | Si |
| 2 | name | VARCHAR () | Nombre del servidor asignado por el usuario. |  | Xxxxxxx | Si |
| 3 | host | VARCHAR () | Nombre del hosting puede ser con dirección Ip. |  | 255.255.255.255 | Si |
| 4 | port | INT | Puerto del servidor definido en 3306. |  | 9999 | Si |
| 5 | username | VARCHAR () | Nombre de usuario en el servidor remoto. |  | Xxxxxxx | Si |
| 6 | password | VARCHAR () | Contraseña de usuario en el servidor remoto. |  | Xxxxxxx | Si |
| 7 | database | VARCHAR () | Base de datos alojada en el servidor remoto. |  | Xxxxxxx |  |
| 8 | description | VARCHAR () | Descripción del servidor. |  | Xxxxxxx |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la Tabla | | Sentence | | | | |
| Nombre de los Campos | | **Tipo de Dato** | **Contenido** | **PK FK** | **Formato** | **Requerida** |
| 1 | idsentence | INT | Id de la sentencia. | PK | 9999 | Si |
| 2 | idserver | INT | Id del servidor. | FK | 9999 | Si |
| 3 | name | VARCHAR () | Nombre definido por el usuario. |  | Xxxxxxx | Si |
| 4 | typesentence | BOOLEAN | Tipo de sentencia SQL o de graficación. |  | True o false | Si |
| 5 | query | VARCHAR () | Sentencia guardada por el usuario. |  | Xxxxxxx | Si |
| 6 | description | VARCHAR () | Descripción de la sentencia. |  | Xxxxxxx |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la Tabla | | Graphic | | | | |
| Nombre de los Campos | | **Tipo de Dato** | **Contenido** | **PK FK** | **Formato** | **Requerida** |
| 1 | idgraphic | INT | Id de la gráfica. | PK | 9999 | Si |
| 2 | idsentence | INT | Id de la sentencia. | FK | 9999 | Si |
| 3 | name | VARCHAR () | Nombre de la gráfica asignada por el usuario. |  | Xxxxxxx | Si |
| 4 | file | VARCHAR () | Ruta del fichero donde localiza la gráfica. |  | Xxxxxxx | Si |
| 5 | description | VARCHAR () | Descripción de la gráfica. |  | Xxxxxxx |  |

## Diseño Visual Y Secuencia De Gestos

La interface es la capa entre el usuario y el corazón funcional de la aplicación es donde nacen las interacciones. En mayor medida está compuesta por botones, gráficos, iconos y fondos que tiene una apariencia visual diferente en cada uno de los sistemas operativos, ya que Android, iOs y Windows Phone tienen su propio estándar de diseño.

### Wireframe De La Actividad De Inicio

El wireframe de inicio está compuesto por una barra de actividades y una lista de los diferentes servidores remotos a los que se tienes acceso. Al realizar una pulsación larga en uno de los ítems de la lista se podrá realizar las acciones de eliminar y editar. La figura de la izquierda, es la interfaz propuesta, mientras que la de la derecha, es la interfaz final de la aplicación, posteriormente se sustituyó la flecha de edición y eliminación por un menú contextual.

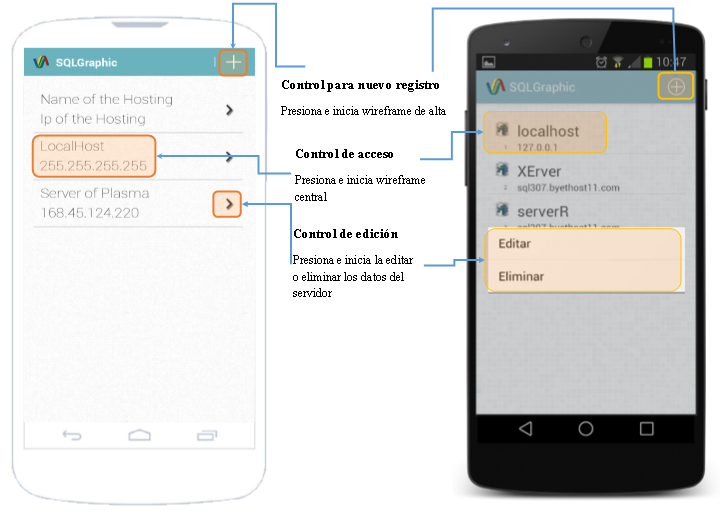


Figura 5. - Wireframe de inicio de SQLGraphic.

### Wireframe De La Actividad De Altas

En la figura 5.4 se aprecia la interfaz que tendrá el wireframe de alta, constituidos por campos de texto para los datos del servidor como, nombre, host, puerto, nombre del usuario, contraseña de acceso, base de datos y descripción del servidor. La barra de acción está formada por el logo de la aplicación, un botón de test de conexión y otro para guardar los datos en la base local. El diseño final es completamente semejante al diseño propuesto para este módulo.

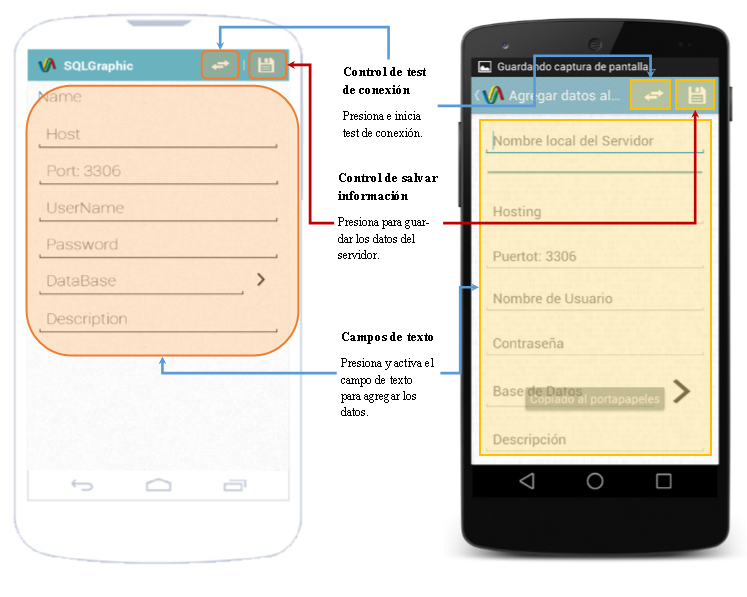


Figura 5. - Wireframe de alta de la aplicación SQLGraphic.

### Wireframe De La Actividad Central

El wireframe central se lleva a cabo la composición de sentencias y el tipo, la barra de acción está compuesta por un botón para guardar la sentencia redactada, otro que ejecuta la sentencia y uno menú desplegable que se modificó en el diseño final por un menú lateral. El menú está compuesto por una lista, que contiene un cierre de conexión, las sentencias guardadas, una galería de graficas almacenas e información sobre la aplicación. La parte de en medio la forma un spinner, una lista de radio botones, un campo de texto para la sentencia y un tab que funciona como auxiliar de redacción.

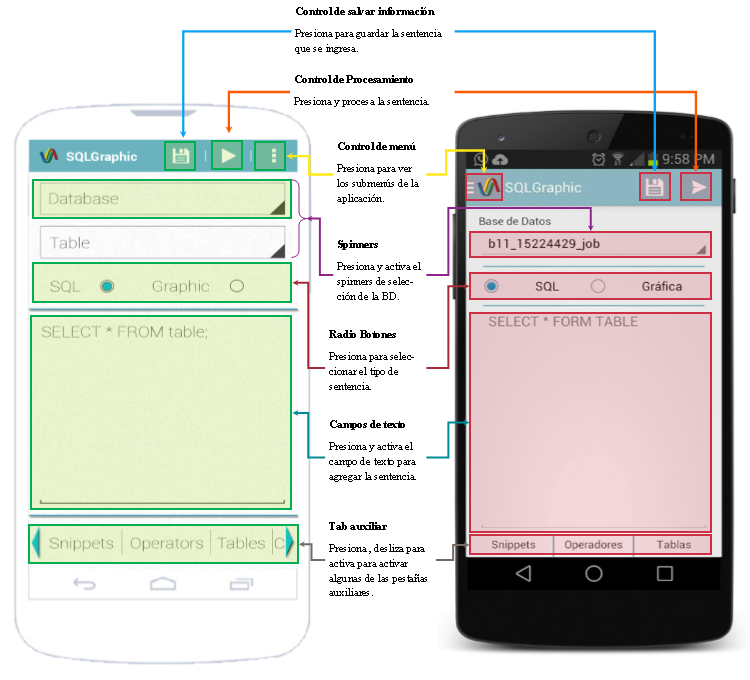


Figura 5. - Wireframe central de la aplicación SQLGraphic.

### Wireframe De La Actividad De Tablas De Resultados SQL

En la figura 5.6 se aprecia el diseño de la interfaz de resultados de la sentencia de tipo tabla, conformada por la barra de acción que a su vez la integran dos botones, uno para guardar la sentencia que genero la tabla y el otro para actualizar la tabla de resultados. En la barra inferior se creó un control de páginas para poder manipular la visualización de la información. La tabla será completamente dinámica depende del resultado de la sentencia, el total máximo de filas mostradas a pantalla es de 30, debido a la dimensión que presentan las pantallas de los dispositivos móviles. El diseño final cumple en la totalidad con el diseño propuesto.

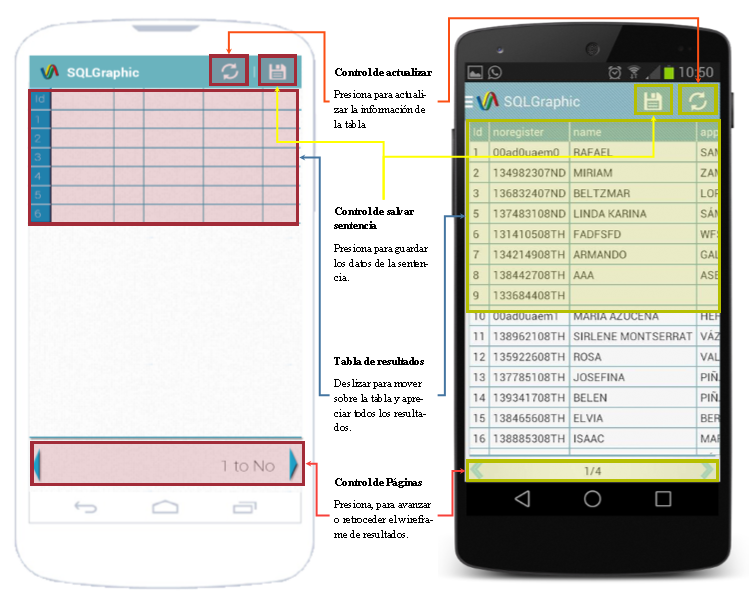


Figura 5. - Wireframe de resultados SQL de la aplicación SQLGraphic.

### Wireframe De La Actividad De Graficación

El wireframe muestra la gráfica de resultados de la sentencia de graficación, está integrada por la barrar de acción, dicha barra la componen los botones de guardar y de actualizar, el botón de actualizar fue sustituido en la interfaz final por un botón de configuración como se aprecia en la figura inferior. La mayoría de la pantalla se ve cubierta por la gráfica creada a partir columna que selecciono el usuario.

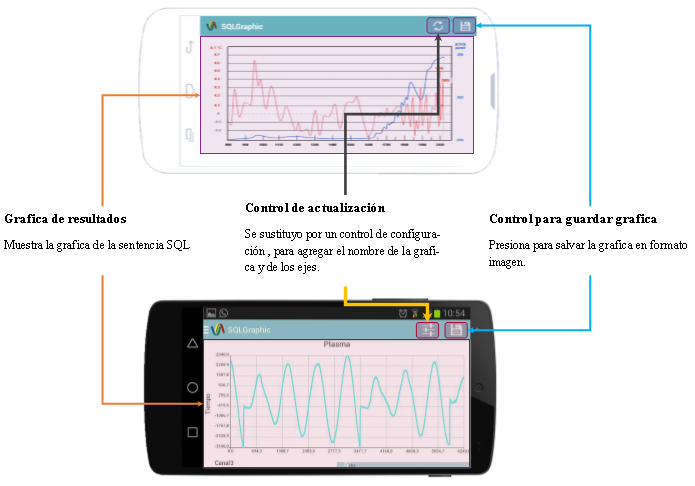


Figura 5. - Wireframe de graficación de la aplicación SQLGraphic.

# Capítulo VI - Pruebas De La Aplicación

## Pruebas De Software

Las pruebas de sistema, se realizan al finalizar el análisis; el diseño; y la construcción global del sistema. Dichas pruebas tienen el propósito de comprobar la funcionalidad e integridad global del sistema, en un entorno lo más parecido posible al entorno final del producto. El proceso de ejecución de pruebas de sistema debe ser considerado en todo el ciclo de vida de un proyecto para así obtener un producto de alta calidad. El éxito, de esas pruebas, dependerá del seguimiento de una estrategia de pruebas adecuada.

## Plantillas De Pruebas De Software

Las plantillas para casos de pruebas es un formulario con campos marcado, y sirve para mejorar los casos de prueba paso a paso, es decir, permite darle orden a los casos de prueba. De esta forma, es posible observar el proceso impreso y ayudar a los probadores a encontrar información. Una vez definido el concepto que nos ocupa se procede a desarrollar las Plantillas de casos de prueba de software, de las cuales debido a la extensión del documento, solo se incluyeron las figuras más significativas.

### Plantilla Para Agregar Datos Del SMDB

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador**: Requisitos Funcionales 1,2 y 3 | | | **Versión**:1.0 |
| **Nombre del proyecto**: SQLGraphic | | |
| **Caso No**: 1 | | | **Ejecución No**: 7 |
| **Nombre del Caso**: | |  | **Nombre**: Agregar datos del servidor |
| Agregar datos del server | | | **Estado de la prueba**: Terminado |
| **Modulo**:  AddServerData.java | | | **Dependencia**:  Base de datos interna  Test de conexión |
| **Escrito por**: Rafael Samano Pichardo | | | **Ejecutado por**: Rafael Samano Pichardo |
| **Fecha**: 03/11/14 | | | **Fecha**:04/11/14 |
| **Propósito**: | | | |
| Comprobar que el modulo permite agregar correctamente los datos del servidor externo. Además de que permita realizar pruebas de conexión al servidor externo y poder gradar los datos en la base interna de la aplicación móvil. | | | |
| **Ambiente de la prueba/Configuración** | | | |
| **Hardware** | Pantalla AMOLED (800 x 480) 4 pulgadas (233ppi).  Procesador 1 GHz. Doble núcleo ARM Cortex-A9  Memoria RAM 1 Gb.   Tarjeta SD de Gb. | | |
| **Software** | Sistema Operativo Android 4.0 Ice Cream Switch. | | |
| **Datos de entrada** | Datos necesarios:   * Nombre del servidor. * Hosting. * Puerto: 3306. * Nombre de usuario.   Datos no necesarios:   * Contraseña. * Base de datos. * Descripción. | | |
| **Pre-requisitos** | | | |
| * El servidor externo como el de la aplicación deben de permanecer online. * El dispositivo móvil requiere contar con una conexión a internet. * El dispositivo necesita contar con una memoria SD. * El dispositivo cuente con 5 Mb mínimo de espacio libre. | | | |
| **Acciones** | | | |
| 1. Cambiar de la actividad Inicial a la actividad de agregar servidor. 2. Ingresar nombre del servidor local. 3. Ingresar nombre del host, puede ser por la Ip del server o por el nombre de dominio del servidor de Mysql. 4. Ingresar puerto de conexión. En caso de no conocer el puerto de conexión definirlo en 3306. 5. Ingresar nombre de usuario. 6. Ingresar la contraseña para el usuario anterior. 7. Clic en el botón conexión que se encuentra en la barra de acción.  * Si alguno de los campos requeridos no se insertaron, la aplicación genera un mensaje de error. * En caso contrario se ejecutara una prueba de conexión. * Si la prueba de conexión es exitosa mandara un cuadro de dialogo con la confirmación de la acción. * De lo contrario genera un mensaje equívoco de conexión. * Al logra conectarse al servidor se podrá seleccionar una de las bases de datos que se hallan alojadas en él.  1. Seleccionar base de datos:  * Clic en el botón lateral al campo de base de datos. * Seleccionar una de las bases de datos que saldrán en un mensaje en forma de lista.  1. Insertar una descripción breve del servidor o de la base de datos a la cual se tiene acceso. Sea que se pueda o no conectar al servidor es posible guardar los datos del servidor externo, al dar clic en el botos de guardar en la barra de acciones. Siempre y cuando se agreguen los campos necesarios. 2. Al guardar los datos del servidor externo en la base de datos local se manda un cuadro de dialogo con la confirmación o el error de la acción.  * Si el mensaje es validado el éxito de la tarea, se prosigue a dar clic en “OK” y automáticamente se limpia el formulario.  1. Cambiar de la actividad de agregar datos del servidor a la actividad inicial para ver la lista actualizada o ingresar los datos de un nuevo servidor externo. | | | |
| **Finalización** | | | |
| * La base de datos de la aplicación se modificará al ser agregado los datos del servidor externo. * La lista de la actividad inicial se actualizara. | | | |

Es de aclararse que, en la primera prueba realizada, no se agregó información al campo del nombre local, hosting, puerto, usuario y contraseña para verificar la validación de campos requeridos. Por lo tanto, se comprobó el funcionamiento de la aplicación como se muestra en la siguiente tabla.

Tabla de Caso de Prueba -

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | No se ingresaron campos requeridos. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de valores requeridos para realizar conexión. | Correcto. |

Como se advierte de la anterior tabla, la prueba generó la salida esperada a la acción que realiza el usuario conforme a la plantilla propuesta, pero el mensaje obtenido no especifica que campo o campos son los que se requieren. Posteriormente se agregan los datos de un servidor externo inexistente para comprobar la conexión, como se aprecia en la tabla de prueba 2:

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se ingresaron campos requeridos de un servidor externo irreal. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de conexión errónea. | Correcto. |

La prueba previa de conexión ratifica que no se puede establecer una conexión con un servidor inexistente. De forma posterior, se realizó otra prueba misma en la que se añadió la información de un servidor que bloquea el acceso a la información que lo conforma, como lo ejemplifica la tabla de prueba 3.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se ingresaron campos requeridos de un servidor externo bloqueado. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de conexión exitosa. | Fallido. |

La prueba en comento resulto fallida a causa de que la aplicación móvil no hace distinción alguna entre un servidor bloqueado y un servidor que no permite el acceso remoto. Para finalizar las pruebas de conexión, se completaron los campos requeridos por la aplicación con información de un servidor externo, para verificar que el sistema puede conectarse exitosamente con el servidor, como se muestra en la tabla de caso de prueba siguiente:

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se ingresaron campos requeridos de un servidor externo. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de conexión exitosa. | Correcto. |

Por otro lado, en la tabla de prueba 5, se intenta seleccionar una base de datos sin realizar una conexión previa, a efecto de corroborar si al añadir los datos del servidor se puede tener acceso a las bases de datos que lo integra, tal y como se muestra a continuación:

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se ingresaron campos requeridos de un servidor externo. | -------- | Correcto |
| 4 | Clic en el botón lateral de base de datos. | Mensaje con la lista de las base de datos. | Fallido. |

El resultado de la prueba anterior, se considera fallido debido a que es necesario realizar una prueba de conexión posterior a la adición de los siguientes campos: hosting, puerto, usuario y contraseña del servidor. Siendo estos últimos los que nos permiten tener acceso al listado de bases de datos que los conforman.

Por otra parte, en la tabla de pruebas 6 se giró el dispositivo móvil al realizar la prueba de conexión con el servidor, esto con la finalidad de garantizar que la aplicación no tiene conflictos con el sensor del dispositivo.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se ingresaron información de un servidor externo. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de conexión exitosa. | Fallido. |

De esta prueba podemos apreciar que al rotar el dispositivo móvil mientras se realiza la prueba de conexión, se generó un error en la actividad en proceso, la consecuencia de este error es que la actividad en curso se cierra después de enviar un mensaje a pantalla y retornado a la actividad inicial de la aplicación móvil. Se efectuaron las acciones lo más apegado a la plantilla de pruebas aunque se obtuvo un resultado correcto, no se considera apropiado tener que comprobar la conexión con el servidor para poder seleccionar la base de datos. Los resultados del seguimiento de las acciones esta prueba se muestra en la tabla de casos de prueba 7 y ejemplificado en la figura 6.2.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en botón agregar. | Inicia la actividad de agregar datos. | Correcto |
| 2 | Se añade la información del servidor en los campos requeridos. | -------- | Correcto |
| 3 | Clic en el botón de conexión. | Mensaje de conexión exitosa. | Correcto. |
| 4 | Clic en el botón lateral de base de datos. Seleccionar base de datos. | Mensaje con las listas de la base de datos y se añade la base seleccionada. | Correcto. |
| 5 | Se ingresa una descripción del sistema gestor. | -------- | Correcto. |
| 6 | Clic en el botón de guardar datos en la base local de la aplicación. | Mensaje que confirma que los datos se agregaron a la base local. | Correcto. |
| 7 | Clic en el botón para retroceder a la actividad de origen. | Se actualiza la lista en la actividad inicial. | Correcto. |

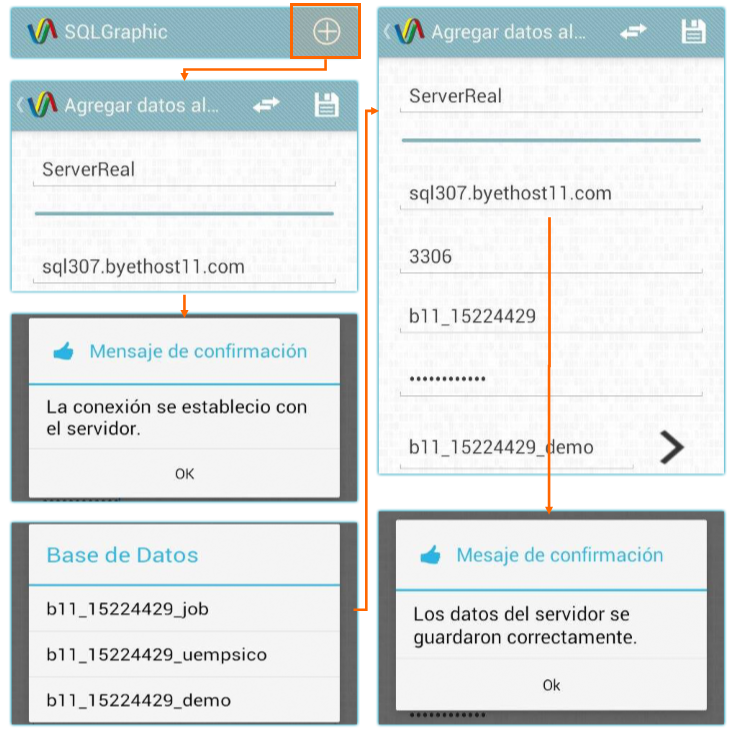


Figura 6. – Resultado de la tabla de caso de prueba 7

Con base en los resultados anteriores, se propone especificar que campos de texto que se requieren para realizar el test de conexión. Además se estima innecesario dar clic al botón de conexión para poder seleccionar la base de datos. Finalmente se concluye que el sistema debe reconocer cuando un servidor se encuentra bloqueado, lo cual implica que se pueda ingresar la base de datos de forma manual.

### Plantilla De Editar Y Eliminar Datos

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador**: Requisitos Funcionales 1,2 y 3 | | | **Versión**:1.0 |
| **Nombre del proyecto**: SQLGraphic | | |
| **Caso No**: 2 | | | **Ejecución No**: 9 |
| **Nombre del Caso**: | |  | **Nombre**: Eliminar y editar datos del servidor |
| Eliminar y editar datos del servidor | | | **Estado de la prueba**: Terminado |
| **Modulo**:  EditActivity.java y StartActivity.java | | | **Dependencia**:  Base de datos interna  Test de conexión |
| **Escrito por**: Rafael Samano Pichardo | | | **Ejecutado por**: Rafael Samano Pichardo |
| **Fecha**: 03/11/14 | | | **Fecha**:05/11/14 |
| **Propósito**: | | | |
| Demostrar que el modulo permite editar y eliminar correctamente la información del servidor externo de la lista y la base de datos de la aplicación. Para editar los datos se ingresa a una actividad nueva, mientras que para eliminar los datos se realiza en la actividad inicial. | | | |
| **Ambiente de la prueba/Configuración** | | | |
| **Hardware** | Pantalla AMOLED (800 x 480) 4 pulgadas (233ppi).  Procesador 1 GHz. Doble núcleo ARM Cortex-A9  Memoria RAM 1 Gb.   Tarjeta SD de Gb. | | |
| **Software** | Sistema Operativo Android 4.0 Ice Cream Sandwich. | | |
| **Datos de entrada** | Datos necesarios recuperados de la base de datos de la aplicación:   * Nombre del servidor. * Hosting. * Puerto: 3306. * Nombre de usuario.   Datos no necesarios recuperados de la base de datos de la aplicación:   * Contraseña. * Base de datos. * Descripción. | | |
| **Pre-requisitos** | | | |
| * El servidor externo como el de la aplicación deben de permanecer online. * El dispositivo móvil requiere contar con una conexión a internet. * El dispositivo necesita contar con una memoria SD. * El dispositivo cuente con 5 Mb mínimo de espacio libre. * La información del servidor externo debe de encontrarse alojada en la base local. | | | |
| **Acciones para editar datos** | | | |
| 1. Realizar una pulsación larga en el ítem de la lista inicial, de esta acción se despliega un menú contextual con las opciones de editar o eliminar, seleccionar la opción de editar. Cambia automáticamente de la actividad inicial a la actividad de editar datos del servidor. 2. Editar los campos que se sean requieran actualización, recuerde que existen campos de texto que no pueden quedar vacíos. 3. Clic en el botón conexión que se encuentra en la barra de acción, para verificar el enlace directo con el servidor externo.  * Si alguno de los campos requeridos no se insertaron, la aplicación generara un mensaje de error. * En caso en caso contrario se ejecutara una prueba de conexión. * Si la prueba de conexión es exitosa mandara un cuadro de dialogo con la confirmación del enlace. * De lo contrario mandara un mensaje erróneo de conexión. * Al logra conectarse al servidor se podrá seleccionar una de las bases de datos que se hallan alojadas en él.  1. Para editar la base de datos seleccionada:  * Clic en el botón que se encuentra adyacente al campo de base de datos. * Seleccionar una de las bases de datos que se despliegan en un mensaje en forma de lista.  1. Sea que se pueda o no conectar al servidor es posible actualizar los datos del servidor externo, al dar clic en el botos de actualizar en la barra de acciones. Siempre los campos que son requeridos por la actividad no se encuentren vacíos. 2. Al actualizar la información del servidor externo en la base de datos local se manda un cuadro de dialogo con la confirmación o el error en la acción.  * Se despliega un mensaje de éxito, se prosigue a dar clic en “OK” y el formulario conservara los datos ya actualizados.  1. Cambiar de la actividad de editar datos del servidor a la actividad inicial para ver la lista actualizada.   **Acciones para eliminar datos**   1. Realizar una pulsación larga en el ítem de la lista inicial, después se despliega un menú contextual con las opciones de editar o eliminar, seleccionar la opción de eliminar. 2. Si se realiza correctamente la acción para eliminar el registro, se envía un mensaje de confirmación de que fue eliminada la información del servidor de la base local. 3. Clic en el botón de “OK” para cerrar el mensaje, la lista se actualiza automáticamente. | | | |
| **Finalización**   * La base de datos de la aplicación se modificara al ser editados o eliminados los datos del servidor externo. * La lista de la actividad inicial se actualiza. | | | |

Por lo que respecta a la plantilla de editar y eliminar datos, tenemos que la prueba se inicia con el módulo de edición. Bajo tal premisa, se comenzó a probar la aplicación eliminando los campos que constituyen la información del servidor externo en la base local, con el objetivo de realizar un chequeo de conexión, tal y como se aprecia en la tabla de caso de prueba 8:

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Pulsación larga sobre la lista de inicial y seleccionamos “Editar”. | Inicia la actividad de editar datos del servidor. | Correcto |
| 2 | Se elimina la información del formulario. | El nombre local de la base de datos no se puede modificar. | Fallido |
| 3 | Clic en el botón de conexión. | Mensaje de conexión exitosa. | Correcto. |

Parte de la prueba anterior resulto fallida debido a que se consiguió modificar el nombre que identifica al servidor en la base de datos local de la aplicación, en cuanto a las otras acciones, la actividad informó que algunos campos requeridos se encontraban vacíos. Posteriormente se realizaron tres pruebas diferentes en dónde; la primera de ellas implicaba modificar la información del servidor actual por la de un servidor inexistente; la segunda donde la información fue sustituida nuevamente por la de otro servidor con el acceso bloqueado; y una última en la cual se cambió la información por la de un servidor sin bloquear. Es oportuno mencionar que en cada una de las iteraciones se hicieron conexiones externas y los resultados obtenidos fueron semejantes a las pruebas de la plantilla anterior. Dichas semejanzas consistieron en que. Hecho lo anterior se modificó la base de datos y no realizar ninguna prueba de conexión previa, obteniendo el siguiente resultado:

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Pulsación larga sobre la lista de inicial y seleccionamos “Editar”. | Inicia la actividad de editar datos del servidor. | Correcto |
| 2 | No se editó la información del servidor. | ------- | Correcto |
| 4 | Clic en el botón lateral al campo de la base de datos. | Mensaje con la lista de las base de datos. | Fallido. |

En la tabla anterior se pudo apreciar que la prueba falló a causa de que el cuadro de dialogo desplegado a pantalla solicita que se realice una prueba de conexión previa para obtener las base de datos que contiene, aun cuando los campos de texto necesarios para realizar la conexión se encuentran añadidos. Posteriormente se editó la información del servidor sin modificar la base de datos y se procedió a actualizar el registro sin perpetrar una prueba de conexión. A efecto de ilustrar la prueba en mención, la tabla de Caso de Prueba muestra el resultado de ésta.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Pulsación larga sobre la lista de inicial y seleccionamos “Editar”. | Inicia la actividad de editar datos del servidor. | Correcto |
| 2 | Se editó la información del formulario. | El nombre local de la base de datos no se puede modificar. | Fallido |
| 5-6 | Clic en el botón de actualizar datos del servidor externo. | Se despliega un mensaje de error, porque la base de datos no pertenece al servidor actual. | Fallido. |

De acuerdo con la tabla de caso de prueba 10 el fallo se presenta porque el usuario cuenta con acceso para editar un campo que debe estar restringido ya que el sistema no distingue cuando la base de datos no pertenece al servidor nuevo. Al finalizar con el módulo de editar se prosiguió a probar la siguiente sección de la plantilla, que corresponde al módulo con la función de eliminar datos del servidor externo en la base local de la aplicación, dicho modulo se probó y ejecutó correctamente como se muestra en la tabla de caso de prueba 11 y ejemplificado en la figura 6.3.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Pulsación larga sobre la lista de inicial y seleccionar “Eliminar”. | Se eliminan los datos de la base local. Se despliega un mensaje. | Correcto |
| 2-3 | Clic en el botón de “OK” | Se actualiza la lista inicial de la actividad | Correcto |

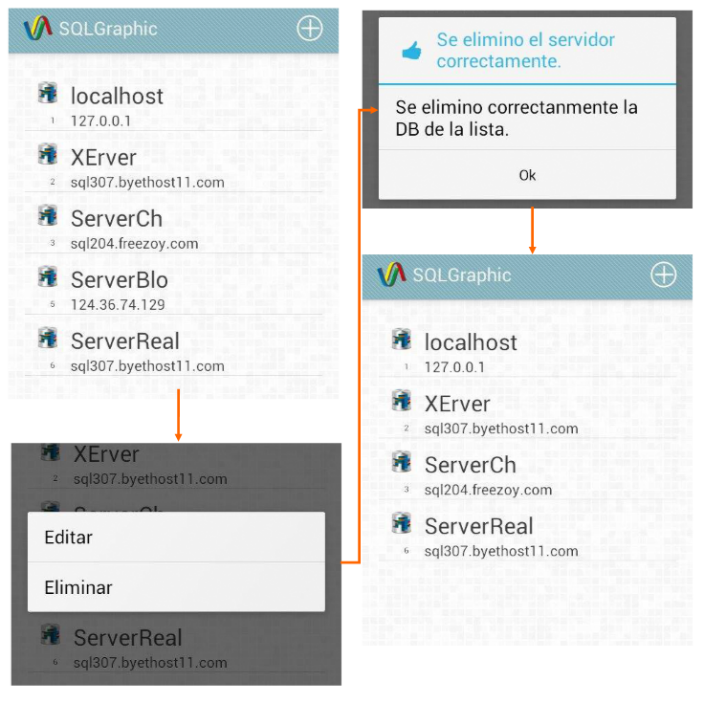


Figura 6. 2 - Resultado de la tabla de caso de prueba 11.

En base a las pruebas previas en el módulo de edición, se sugiere mejorar la estructura para la edición de datos, así mismo, lograr que identifique cuando la base de datos no pertenece al servidor editado.

### Plantilla De La Actividad Central

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador**: Requisitos Funcionales 4,5 | | | **Versión**:1.0 |
| **Nombre del proyecto**: SQLGraphic | | |
| **Caso No**: 3 | | | **Ejecución No**: 6 |
| **Nombre del Caso**: | |  | **Nombre**: Consultas SQL o Graficación |
| Consultas SQL o Graficación | | | **Estado de la prueba**: Terminado |
| **Modulo**:  MainActivity.java y Fragment\_Query.java | | | **Dependencia**:  Base de datos interna  Test de conexión  Estructura información del servidor externo |
| **Escrito por**: Rafael Samano Pichardo | | | **Ejecutado por**: Rafael Samano Pichardo |
| **Fecha**: 03/11/14 | | | **Fecha**:06/11/14 |
| **Propósito**: | | | |
| Corroborar que el modulo en cuestión funcione correctamente, al permitir seleccionar alguna de las base de datos disponibles para el usuario, elegir alguna de las dos formas globales de trabajo que son tabla (sentencia SQL) o graficación. También de poder construir la sentencia SQL a través de los botones auxiliares (Snippet, operadores y tablas). | | | |
| **Ambiente de la prueba/Configuración** | | | |
| **Hardware** | Pantalla AMOLED (800 x 480) 4 pulgadas (233ppi).  Procesador 1 GHz. Doble núcleo ARM Cortex-A9  Memoria RAM 1 Gb.   Tarjeta SD de Gb. | | |
| **Software** | Sistema Operativo Android 4.0 Ice Cream Sandwich. | | |
| **Datos de entrada** | Datos recuperados de la base de datos de la aplicación:   * Nombre del servidor. * Hosting. * Puerto: 3306. * Nombre de usuario. * Contraseña.   Datos requeridos   * Base de datos. * Tipo de Sentencia. * Sentencia SQL.   Datos requerido para guardar sentencia   * Nombre de la sentencia. * Descripción de la sentencia. * Sentencia SQL. | | |
| **Pre-requisitos** | | | |
| * El servidor externo como el de la aplicación deben de permanecer online. * El dispositivo móvil requiere contar con una conexión a internet. * El dispositivo necesita contar con una memoria SD. * El dispositivo cuente con 5 Mb mínimo de espacio libre. * La información del servidor externo debe de encontrarse alojada en la base local. | | | |
| **Acciones** | | | |
| 1. Realizar una pulsación en el ítem del servidor al cual se dese conectar, posteriormente la aplicación tratara de conectarse al servidor externo de manera directa.  * En caso de que no se pueda establecer conexión con el servidor externo se despliega un mensaje de error de conexión. * De lo contrario se inicia la actividad de consultas y se carga la estructura con las dase de datos y las tablas que conforman al servidor.  1. Seleccionar las base de datos en donde se pretende ejecutar la sentencia SQL o de graficación. 2. Elegir alguna de las dos clases sentencias:  * Tabla – genera una tabla a partir de una sentencia de SQL. * Graficación – genera una gráfica dinámica en base de la sentencia de SQL.  1. Ingresar la sentencia SQL en el formato adecuado.  * Se puede auxiliar de los botones que se encuentran en la parte inferior de la pantalla para crear las sentencias SQL. Los cuales están integrados por palabras reservadas, operadores y tablas que varían de acuerdo a la base de datos seleccionada.  1. Inmediatamente después de crear la sentencia SQL se puede ejecutar en el servidor externo al dar clic en el botón “Ejecutar”, que está situado en la barra de acciones.  * Si la sentencia SQL no se encuentra determinada, la aplicación mandara un mensaje de error. * Si se cumplen los requerimientos el sistema verificara el tipo de sentencia, si es de tipo tabla cambia a la actividad de tabulación, en caso contrario inicia la actividad de graficación. * Se crea la estructura con la información necesaria para crearla tabla o la gráfica dado el caso.   **Acciones para guardar sentencia**   1. Después de seleccionar la base de datos y de redactar la sentencia se puede guardar la sentencia SQL al dar clic en el botón de guardar.  * Ingresar nombre de la sentencia. * Ingresar descripción de la sentencia. * La sentencia SQL se agregar automáticamente.  1. Si se agrega correctamente la sentencia en la tabla se manda un mensaje a pantalla de confirmación, de lo contrario se manda un mensaje de error. | | | |
| **Finalización** | | | |
| * Se genera la estructura de información con el id los datos, la base de datos y la sentencia SQL. * Guardar la sentencia SQL que se genera en la interfaz en la base local de la aplicación. | | | |

Por lo que se refiere a la plantilla central, se tiene que cada una de las pruebas se inicializan en la lista de la actividad inicial, dicho lo anterior se prosigue a seleccionar uno de los ítems de la referente lista el cual pertenece a un servidor fuera de línea, con el propósito de, confirmar que la aplicación no ingresa a la actividad central sin tener conexión. Como resultado de la prueba se corrobora que la aplicación móvil trata de conectarse al servidor externo, sin no tener éxito despliega un mensaje a pantalla de error de conexión, como se aprecia en la siguiente tabla.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Se pulsa alguno de los ítems de la lista inicial. | Se despliega un mensaje de error, si el servidor no tiene acceso remoto. | Correcto |

Por el contrario, en la siguiente prueba se seleccionó un ítem que corresponde a un servidor con conexión remota, después de lograr la conexión se inicia la actividad central. Posteriormente al analizar la interfaz con perspicacia, resulta que la base de datos que se guardó previamente en la base local no aparece al inicio de la lista, como se ejemplifica en la tabla de caso de prueba 13.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Se pulsa el ítem correspondiente al servidor con el que se va a trabajar. | Inicia la actividad de central y se crea la estructura de base de datos y tablas | Correcto |
| 2 | Se pulso sobre el spinner de base de datos para seleccionar la base de datos con la que se trabaja. | El listado de base de datos debe de iniciar con la base de datos guardada en la base local. | Fallido |
| 3 | Clic en el radio botón del tipo de sentencia a ejecutar. | Se selecciona la el tipo de sentencia. | Fallido |

Para terminar la prueba se optó por uno de los dos tipos de sentencia, al realizar acción anterior la actividad en curso se detiene, como consecuencia: se desplegar un mensaje error a pantalla, y el sistema retrocede a la actividad previa. Es de mencionar que se solucionó el error correspondiente a la selección del tipo de sentencia, con el propósito de proseguir con la pruebas, después de la corrección el sistema respondió de forma adecuada, como se muestra en la siguiente tabla de caso prueba.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Se pulsa el ítem correspondiente al servidor a usar. | Inicia la actividad de central y se crea la estructura de base de datos y tablas | Correcto. |
| 2 | Se pulso sobre el spinner de base de datos. | Aparece el listado de base de datos que forman el servido. | Correcto. |
| 3 | Clic en el radio botón del tipo de sentencia a ejecutar | Se selecciona la el tipo de sentencia. | Correcto. |
| 4 | Se redacta la sentencia SQL con ayuda de la barra inferior. | Al usar el auxiliar de la barra inferior se despliega una lista referente al botón presionado. | Correcto. |
| 5 | Clic en el botón de ejecutar sentencia. | Se inicializa la actividad dependiendo del tipo de sentencia elegida. | Correcto. |

### Platilla De Tabla Sql

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador**: Requisitos Funcionales 5 y 6 | | | **Versión**:1.0 |
| **Nombre del proyecto**: SQLGraphic | | |
| **Caso No**: 4 | | | **Ejecución No**: 5 |
| **Nombre del Caso**: | |  | **Nombre**: Tabla SQL |
| Tabla SQL | | | **Estado de la prueba**: Terminado |
| **Modulo**:  Fragment\_Table.java | | | **Dependencia**:  Base de datos interna  Test de conexión  Sentencia SQL  Base de datos externa |
| **Escrito por**: Rafael Samano Pichardo | | | **Ejecutado por**: Rafael Samano Pichardo |
| **Fecha**: 03/11/14 | | | **Fecha**:10/11/14 |
| **Propósito**: | | | |
| Verificar que el sistema genere una estructura dinámica de páginas de acuerdo al número de registros concebidos por la sentencia SQL. Que los objetos para navegar de forma más ágil a través de las páginas de la actividad realicen su función de manera puntual. | | | |
| **Ambiente de la prueba/Configuración** | | | |
| **Hardware** | Pantalla AMOLED (800 x 480) 4 pulgadas (233ppi).  Procesador 1 GHz. Doble núcleo ARM Cortex-A9  Memoria RAM 1 Gb.   Tarjeta SD de Gb. | | |
| **Software** | Sistema Operativo Android 4.0 Ice Cream Sandwich. | | |
| **Datos de entrada** | Datos recuperados de la base de datos de la aplicación:   * Nombre del servidor. * Hosting. * Puerto: 3306. * Nombre de usuario. * Contraseña.   Datos requeridos   * Base de datos. * Tipo de Sentencia. * Sentencia SQL.   Datos requerido para guardar sentencia   * Nombre de la sentencia. * Descripción de la sentencia. * Sentencia SQL. | | |
| **Pre-requisitos** | | | |
| * El servidor externo como el de la aplicación deben de permanecer online. * El dispositivo móvil requiere contar con una conexión a internet. * El dispositivo necesita contar con una memoria SD. * El dispositivo cuente con 5 Mb mínimo de espacio libre. * La información del servidor externo debe de encontrarse alojada en la base local. | | | |
| **Acciones** | | | |
| 1. Inicia previamente al dar clic en el botón de ejecutar de la actividad de consulta. El sistema verifica el tipo sentencia y ejecuta la actividad correspondiente. 2. Al iniciar la actividad se ejecuta un método asíncrono que realiza la conexión con el servidor externo.  * Si el dispositivo no cuenta con conexión a internet, el sistema despliega un mensaje de error y cambia a la actividad anterior. * Si la conexión a internet es correcta, pero la sentencia SQL no se puede procesar por: errores de sintaxis, servidor incorrecto o por tiempo de ejecución, despliega un mensaje de error y retornara a la actividad anterior.  1. Al finalizar con la conexión con el servidor y se procesa correctamente la sentencia SQL, el sistema inicia la creación de la interfaz de usuario.  * El número de registros se limita a un número de 30 debido al tamaño de la pantalla.  1. Para avanzar a la página siguiente dar clic en el botón inferior derecho.  * Comprobar que se carguen correctamente cada una de las páginas posteriores.  1. Para retroceder a la página anterior dar clic en el botón inferior izquierdo.  * Verificar que al retroceder en la página 1 no se conciba ninguna clase de error.  1. Para moverse entre las páginas se puede dar clic al numerador central y seleccionar el número de página se desea visualizar.   .  **Acciones para guardar sentencia**   1. Posteriormente puede guardar la sentencia SQL al dar clic en el botón de guardar.  * Ingresar nombre de la sentencia. * Ingresar descripción de la sentencia. * La sentencia SQL se agregar automáticamente.  1. Si se agrega correctamente el registro de la sentencia a la tabla se manda un mensaje a pantalla de confirmación, de lo contrario el mensaje es erróneo.   **Acciones para actualizar sentencia**   1. Para actualizar los datos dar clic en el botón de actualizar en la barra de acciones.  * La acción retorna a la página uno con los datos actualizados en caso de que se exista alguna modificación de los datos. | | | |
| **Finalización** | | | |
| * Se genera una interfaz dinámica de acuerdo al número de columnas que se seleccionan a través de la sentencia SQL. * Poder guardar la sentencia SQL en la base local de la aplicación. * Procesamiento correcto de la sentencia SQL. | | | |

En base a los resultados anteriores, se inicia la prueba estableciendo la premisa de girar el dispositivo móvil mientras realiza la conexión con el servidor externo, con el finalidad de corroborar la inexistencia de algún conflicto con el sensor de rotación del dispositivo móvil, el resultado de las acciones que integra la prueba son expuestas en la tabla de caso de prueba 15.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de ejecutar. | Se inicia la actividad derivada de sentencia de tipo tabla y se genera una conexión al servidor. | Correcto |
| 2 | Se espera a que termine la conexión con el servidor externo y se obtienen los datos. | Se crea la tabla con un número máximo de 30 registros por hoja. | Fallido |

Al concluir con la prueba previa, se corrobora que al girar el dispositivo móvil mientras se realiza la conexión con el servidor se produce un error interno, dicho error detiene la actividad en curso y para después regresar a la actividad preliminar. Por lo tanto, se prosiguió a resolver el conflicto de la conexión con el sensor de rotación y continuar con las pruebas de la aplicación. De forma posterior, se ingresó una sentencia SQL con sintaxis incorrecta, con el objetivo de verificar el funcionamiento de la aplicación al ingresar información errónea, el resultado se visualiza en la siguiente tabla de caso prueba.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de ejecutar. La sentencia esta sintácticamente mal redactada. | Se inicia la actividad derivada de la sentencia de tipo tabla y se despliega a un mensaje a pantalla que no se puede procesar la sentencia. | Correcto |

Después de finalizar la prueba anterior, se inicia una prueba ejecutando las acciones de acuerdo a lo propuesto por la plantilla, con la finalidad de, comprobar si la aplicación es capaz de visualizar el mínimo de registros a pantalla, al mismo tiempo que, se corrobora que las herramientas para avanzar y retroceder las páginas de la tabla funcionen, dicha prueba se observa en la siguiente tabla de caso de prueba.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de ejecutar. | Se inicia la actividad derivada de sentencia de tipo tabla y se genera una conexión al servidor. | Correcto. |
| 2 | Se espera a que termine la conexión con el servidor externo y se obtienen los datos. | Se crea la tabla con un número máximo de 30 registros por hoja. | Correcto. |
| 3 | Clic en el botón de avanzar. | Se muestran 30 registros más de la información global. | Fallido. |
| 4 | Clic en el botón de retroceder. | Se muestran los 30 registros anteriores. | Fallido. |
| 5 | Clic en el numerador de páginas. Elegir la página deseada. | Se muestra el numeral y al seleccionar la página se cargan los 30 registros referentes a la pág. | Fallido. |

En base a los resultados de la tabla de casos de pruebas 17, se genera un error en la aplicación al cargar los datos de las páginas posteriores, los registros 31, 61, 91 y los sucesivos son omitidos por este error. El resultado de esta prueba se puede aprecia en la figura 6.3 que enmarca con un recuadro la acción fallida.

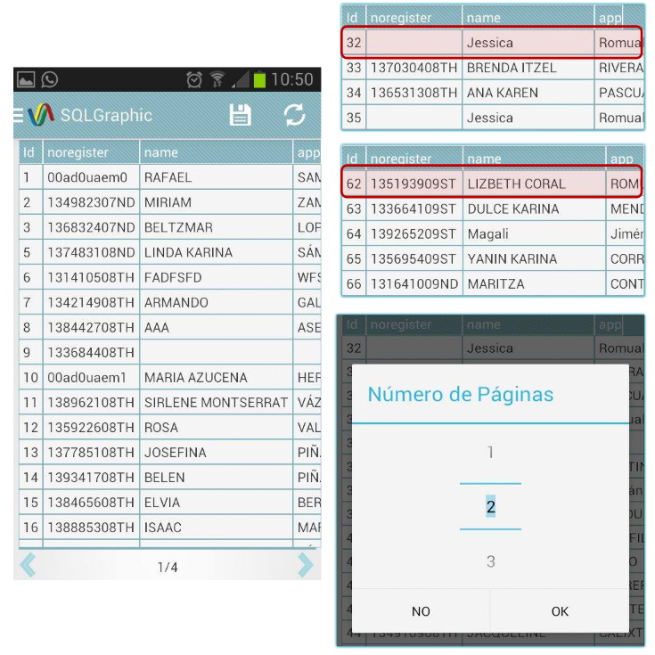


Figura 6. – Resultado de la tabla de caso de prueba 17

Para continuar con las pruebas se intentó guardar la sentencia que es el origen de la tabla que se visualizada en el dispositivo móvil, para esta acción, se inicia dando un clic en el botón de guardar situado en la esquina superior derecha de la pantalla, como resultado de esta acción se despliega un cuadro de dialogo, el cuadro de dialogo se compone de tres campos, específicamente: nombre, descripción y sentencia. Posteriormente, se añade la información solicitada por el cuadro de dialogo, como lo muestra la tabla de caso de prueba posterior.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de guardar. | Se despliega un cuadro de dialogo con campos de texto. | Correcto. |
| 2 | Se añaden los datos requeridos. Después clic en “OK”. | Se genera un mensaje de confirmación. | Correcto. |

Para finalizar con las pruebas, se realizó la acción que implica actualizar la información en la tabla generada a partir de la sentencia SQL, esta acción se inicia al dar clic al botón que se sitúa al lado derecho del botón de guardar, el resultado de la prueba es, que la aplicación no actualiza la tabla a la sección inicial correctamente, como se visualiza en la tabla de caso de prueba 19.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de actualizar. | Se actualiza la tabla en la página inicial. | Fallido. |

Basados en los resultandos referentes a la plantilla de tabla SQL, se opina que la actividad debe de mejorar en: la estructura al guardar los datos de la sentencia especificando a que base de datos pertenece; corregir los bugs que conllevan a la pérdida de un registró al avanzar o retroceder páginas; y por último que el botón de actualizar tabla funcione como se pretende que lo haga.

### Plantilla De Graficar

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador**: Requisitos Funcionales 5 y 7 | | | **Versión**:1.0 |
| **Nombre del proyecto**: SQLGraphic | | |
| **Caso No**: 5 | | | **Ejecución No**: 5 |
| **Nombre del Caso**: | |  | **Nombre**: Graficación de datos |
| Graficación de datos | | | **Estado de la prueba**: Terminado |
| **Modulo**:  Fragment\_Graph.java | | | **Dependencia**:  Base de datos interna  Test de conexión  Sentencia SQL  Base de datos externa |
| **Escrito por**: Rafael Samano Pichardo | | | **Ejecutado por**: Rafael Samano Pichardo |
| **Fecha**: 03/11/14 | | | **Fecha**:17/11/14 |
| **Propósito**: | | | |
| Evaluar que la aplicación desplego una lista de las columnas que integran la sentencia SQL. Además de corroborar que la se grafiquen los datos y se pueda configurar los ejes de la gráfica. | | | |
| **Ambiente de la prueba/Configuración** | | | |
| **Hardware** | Pantalla AMOLED (800 x 480) 4 pulgadas (233ppi).  Procesador 1 GHz. Doble núcleo ARM Cortex-A9  Memoria RAM 1 Gb.   Tarjeta SD de Gb. | | |
| **Software** | Sistema Operativo Android 4.0 Ice Cream Sandwich. | | |
| **Datos de entrada** | Datos recuperados de la base de datos de la aplicación:   * Nombre del servidor. * Hosting. * Puerto: 3306. * Nombre de usuario. * Contraseña.   Datos requeridos   * Base de datos. * Tipo de Sentencia. * Sentencia SQL.   Datos no requeridos para la configuración.   * Nombre de la gráfica. * Nombre del eje de las abscisas. * Nombre del eje de las ordenadas. | | |
| **Pre-requisitos** | | | |
| * El servidor externo como el de la aplicación deben de permanecer online. * El dispositivo móvil requiere contar con una conexión a internet. * El dispositivo necesita contar con una memoria SD. * El dispositivo cuente con 5 Mb mínimo de espacio libre. * La información del servidor externo debe de encontrarse alojada en la base local. | | | |
| **Acciones para graficar** | | | |
| 1. Inicia previamente al dar clic en el botón de ejecutar de la actividad principal. El sistema verifica el tipo de función y ejecuta la actividad correspondiente. 2. El sistema realiza la conexión con el servidor externo usando un método asíncrono.  * Si no tiene conexión el sistema despliega un mensaje de error y regresara a la actividad de consulta. * Si la conexión es correcta pero la sentencia SQL no se puede procesar por: errores de sintaxis, servidor incorrecto o por tiempo de ejecución, manda un mensaje de error y retornara a la actividad anterior.  1. Al finalizar con la conexión y el procesamiento correcto de la sentencia SQL se genera una lista integrada por las columnas de la sentencia SQL, seleccionar la columna que funciona como base de la gráfica. 2. Posteriormente el sistema genera una conexión con el servidor externo, descartando los campos no grafícales y finalizando con la adquisición de los datos por columna en formato JSON.  * En el evento asíncrono el sistema también genera la gráfica. Cotejar tiempos de respuesta para el mejor de graficación de datos.  1. Al finalizar la graficación de los datos el sistema crea la gráfica, para después solicitar la configuración. No es necesario configurar la gráfica en ese momento, ya que la opción se encuentra disponible en el menú de configuración.  * Ingresar nombre de la gráfica. * Ingresar nombre del eje de las abscisas. * Ingresar nombre del eje de las ordenadas. * Seleccionar el tipo de línea: curva o lineal.   **Acciones para guardar imagen de la gráfica.**   1. Al tener la gráfica construida, se cuenta con la opción de guardarla en formato de imagen, para ello se da clic en el botón de guardar que se sitúa en la barra de acciones.  * El sistema verificara si el dispositivo cuenta con una memoria SD, en donde se salvaran la imagen de la gráfica. * En dado caso de que el dispositivo no cuente con dicha memoria el sistema mandara un mensaje en el que se especifica que no se cuenta con una memoria SD. * De lo contrario se guardar la imagen en la memoria SD, específicamente en el fichero “Grafica” destinado para la aplicación.  1. Al terminar se despliega un mensaje de éxito. Se puede acceder a la lista de imágenes en el menú lateral en el apartado de Galería. | | | |
| **Finalización** | | | |
| * Se genera una gráfica dinámica de acuerdo a la columna elegida como base. * Poder guardar la gráfica en forma de imagen en la memoria SD. * Procesamiento correcto de la sentencia SQL. | | | |

Para comenzar con las pruebas con la plantilla de graficar, se contempló previamente que la actividad en curso podría tener conflicto que con el sensor de rotación del dispositivo móvil, con la primicia anterior, se inicia la prueba girando la pantalla del dispositivo mientras se obtienen las columnas que integran la sentencia SQL, esto origina que se retroceda a la actividad previa e inicie el método de conexión nuevamente para obtener las columnas, al finalizar la conexión la aplicación colapsa y retorna a la actividad inicial, como se aprecia en la tabla de caso de prueba 21.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de ejecutar. | Se inicia la actividad derivada de sentencia de tipo gráfica, se inicia una conexión al servidor generando la estructura de columnas. | Correcto |
| 2 | Se espera a que termine la conexión con el servidor externo y se obtienen las columnas de la sentencia. | Se crea una lista de las columnas que genera la sentencia SQL, para después seleccionar la que funciona como base para graficar. | Fallido |

Después de corregir el error que generaba el rotar el dispositivo mientras se lleva a cabo la adquisición de la información, se proseguido con las pruebas correspondientes. La tabla de caso de prueba 22, es una de las tablas que representa una de las diferentes graficas creadas a partir de distintas sentencias SQL(select \* from, entre otras), dichas sentencias varían en cantidad de registros propuestos graficar, al finalizar el proceso de graficar se puede visualizar los datos correctamente graficados, aunque cabe mencionar que el tiempo de espera depende de la cantidad de datos a graficar, ya que mientras mayor sea el número de datos mayor será el tiempo de espera para visualizar el resultado de la gráfica a pantalla. La figura 6.4 ejemplifica las acciones de la tabla de caso de prueba 22.

Tabla de Caso de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| **Paso** | **Acción** | **Salida esperada** | **Resultado** |
| 1 | Clic en el botón de ejecutar. | Se inicia la actividad derivada de sentencia de tipo gráfica, se inicia una conexión al servidor y se crea una estructura dinámica con las columnas. | Correcto. |
| 2 | Se espera a que termine la conexión con el servidor externo y se obtienen las columnas de la sentencia. | Se crea una lista de las columnas que genera la sentencia SQL, para después seleccionar la que funciona como base para graficar. | Correcto. |
| 3 | Clic en el ítem de la columna. Y espera a que finalice la obtención de los datos. | Se despliega un mensaje de carga, que solicita que esperemos a que se genere la gráfica. Posteriormente se crea la gráfica. El tiempo de espera incrementa mientras el número de registros sea mayor. | Correcto. |
| 4 | Se agrega la configuración de la gráfica si se cree pertinente. | Se crea un cuadro de configuración y se ingresan los datos solicitados si el usuario lo requiere. | Correcto. |



Figura 6. – Resultado de la tabla de caso de prueba 22

## Comparativa De SQLGraphic Con Excel

Se confronto las gráficas generadas por la aplicación móvil y el módulo que contiene Excel para realizar gráficas, para corroborar que el graficado utilizado por el dispositivo móvil realiza la gráfica correspondiente a la los datos que se solicitan por medio de la sentencia.

Figura 6. - Serie de plasma graficada en Excel.

## Conclusiones

* Sdadasdas
* Shdghsgdh
* sadasdasdas

# Referencia Bibliográfica

[1] Gil Salvador y Rodríguez Eduardo, “Física re-Creativa. Experimentos de física usando nuevas tecnologías”, Prentice Hall, Buenos Aires, 25 de junio de 2001.

[2] James D. Foley “Computer Graphics: Principles and Practice”, 2da Ed., Addison-Wesley Professional, 1996, 1175 p.

[3] Francis S. Hill, Stephen M. Kelley, “Computer graphics: using OpenGL”, 3ra ed., Pearson Prentice Hall, 2007, 778 p.

[4] James D. Foley, Andries Van Dam, “Fundamentals of interactive computer graphics”, 1ra ed., Addison-Wesley, 1982, 664 p.

[5] TICbeat, "Caen las ventas totales de teléfonos móviles", TICbeat, 14 de Agosto de 2012.

[6] Lara Rodríguez Domingo, “Sistemas Inalámbricos de Comunicación Personal”,1ra ed., México, Marcombo, 2002, 352 p.

[7] Rey Veiga Eugenio, “Telecomunicaciones Móviles”, 2da Ed., Barcelona, España, Marcombo, 1998, 276 p.

[8] Serope Kalpakjian, Steven R. Schmid, Ulises, “Manufactura, ingeniería y tecnología”, 4ta ed., DF, México, Pearson Educación, 2002, 1152 p.

[9] Satya Komatineni, Dave McLean, “Pro Android 4”, 1ra ed., Apress, 2011, 951p.

[10] Hernández Téllez Danny, Gaona García Paulo Alonso, "Prototipo WAP Aplicado a un Ambiente de Aprendizaje virtual (m-learning) con estándares para el desarrollo de Aprendizaje Móvil", Universidad Distrital, Bogotá, Colombia, Eighth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI’2010), “Innovation and Development for the Americas”, June 1-4, 2010, Arequipa, Perú.

[11] Barrón Estrada María Lucia, Zatarain Cabada Ramón, Santillán Hernánde Luis Carlos, "Adaptación Dinámica de Contenidos Educativos en Dispositivos Móviles", Revista Electrónica de Investigación Educativa, Departamento de Sistemas y Computación Instituto Tecnológico de Culiacán, México, 2011.

[12] Gutiérrez Aranzeta, “Introducción a la Metodología Experimental”, 2da ed., DF México, Limusa, 2005, 213 p.

[13] Flores Estrada Claudia, “Las formas básicas de graficación y su relación con situaciones de movimiento”, CECYT, CICATA, IPN, México, 2007.

[14] Torres Bezaury, Araceli María Eugenia, “La modelación y graficación en situaciones de movimiento con tecnología”, XXII Simposio Internacional de Computación en Educación, México, 2006.

[15] Thomas Rist, Patrick Brandmeier, “Customizing Graphics for Tiny Displays of Mobile Devices”, Journal Personal and Ubiquitous Computing, London, Vol. 6, 2002, pp. 260-268.

[16] Capin, T, Dept of Computing Eng., Bilkent Univ., Ankara Pulli, “Computer Graphics and Applications, IEEE”, Vol. 28, IEEE Magazine de IEEE, 2008, pp. 74-84.

[17] W Davis, D Mastrovito, “DbAccess: interactive statistics and graphics for plasma physics databases”, Fusion Engineering and Design, Vol. 71, June 2004, pp. 183-188.

[18] National Instruments, “NI lanza Apps basadas en iOS y Android Mobile para LabVIEW y hardware de NI”, 10 de Abril de 2012.

[19] Nieto González Alejandro, “Las mejores herramientas sobre Matlab para Android”, 4 de Abril de 2012,

[20] Francis F. Chen, “Introduction to Plasma Physics and Controlled Fusion: Plasma physics”, 2da ed., New Cork, EEUU, Plenum Press, 1 Vol 1984, pp. 40-65.

[21] Gordillo Vázquez Francisco José, “Plasmas Fríos”, Revista Investigación y Ciencia, Junio 2008, pp.70-79.

[22] Torres Castro Jesús Pedro, Gamero Rojas Antonio José, Antonio Sola Díaz, “La electricidad y la generación de plasmas”, Revista Anales de mecánica y electricidad, ISSN 0003-2506, Vol. 80, Fasc. 5, 2003, pp. 34-39.

[23] Pallás Areny Ramón “Sensores Acondiciónales de Señal”, 4ta ed., Barcelona, España, Marcombo, 2003, 594 p.

[24] M Prieto, E. Arrieta, "Guerra en el Olimpo de los Móviles", Digitech, Madrid, 01 de junio de 2012, 12p.

[25] Barrón Estrada María Lucia, Zatarain Cabada Ramón, Santillán Hernánde Luis Carlos, "Adaptación Dinámica de Contenidos Educativos en Dispositivos Móviles", Revista Electrónica de Investigación Educativa, Departamento de Sistemas y Computación Instituto Tecnológico de Culiacán, México, 2011, 23p.

[26] A. Kuhn Klaus, R Varren James, Leons Tze-Yun “MedInfo 2007”, 1ra ed., Amsterdam Netherlans, IOS Press, 1 Vol., 1502p.

[27] Maher I. Boulos, Pierre Fauchais, Emil Pfender, “Thermal Plasmas: Fundamentals and applications”, 1ra ed., New York (EEUU), Plenum Press, 1 Vol., 1994, 452 p.

[28] Martínez Bencardino Ciro, “Estadísticas Básicas Aplicadas”, 3ra ed., Bogotá, Colombia, Ecoe Ediciones, 2006, 406p.

[29] Gutiérrez Aranzeta, “Introducción a la Metodología Experimenta”, 2da ed., Distrito Federal, México, Limusa, 2005, 214p.

[30] Vijay Nehra, Ashok Kumar and H K Dwivedi: “Atmospheric Non-Thermal Plasma Sources”, International Journal of Engineering, Vol. 2: Issue, 2008.

[31] Satya Komatineni, Dave MacLean, “Pro Android 4”, 1ra ed., Apress, 2011, 915p.

[32] Reto Merier, “Profesional Android Aplication Development”, 1ra ed., Indiana, EEUU, 2012, 720p.

[33] Sommerville Ian, “Ingeniería en Software”, 7ma ed., Madrid, España, 2005, 712p.

[34] McGlaughlin R, “Software Engineering Notes”, vol. 16, núm. 4, 1991, 53-54 pp.

[35] Naur P., B. Randall (eds.), “Software Engineering: A Report on a Conference Sponsored by the NATO Science Committe”, NATO, 1969.

[36] IEEE Standards Collection: Software Engineering, IEEE Standard 610.12-1990, IEEE, 1993.

[37] S. Pressman Roger, “Ingeniería del Software: Un Enfoque Práctico”, 7ma ed. Distrito Federal, México, Mc Graw Hill, 2010, 767p.

[38] McConnell S., “Software Engineering Priciples”, IEEE Software, vol. 16, núm. 2, 1999.

[39] Loïc Martínez Alonso Fernando, Segovia Javier Fco., "Introducción a la Ingeniería del software: Modelos de desarrollo de programación", 1ra ed., Madrid, España, Delta Publicaciones, 2005, 538p.

[40] Benet Campderrich Falgueras, "Ingeniería del software", 1ra ed., Editorial UOC, 2003, 320p.

[41] Tuya Javier, Ramos Román Isabel, Cosín Dolado Javier, "Técnicas cuantitativas para la gestión en la Ingeniería de Software", 1ra ed., España, Netbiblo, 2007, 373p.

[42] Campderrich Falgueras Beret, “Ingeniería del Software”, 1ra ed. Barcelona, España, Editorial UOC, 2003, 315p.

[43] Humphrey S. Watts, “PSP: A Self-Improvement Process for Software Engineers”, 1ra ed., Massachusetts (EEUU), Person Education, 2005, 210 p.

[44] Rubalcaba Mara, "Procesos de Software", Software Guru Conocimiento en Práctica, No 1, 2005, pp. 20-28.

[45] I. Maletic Jonathan, Howald Anita, Andrian Marcus, “Incorporating PSP into a Traditional Software Engineering Course: An Experience Report”, Proceedings of the Fourteenth Conference on Software Engineering Education and Training, IEEE, 2001.

[46] Stair M. Ralph, Reynolds W. George, “Sistemas de Información: Enfoque Administrativo”, 4ta ed., DF (México), Internacional Thomson Editoriales, 1999, 700p.

[47] Rob Peter, Coronel Carlos, “Sistemas de Base de Datos: Diseño, implementación y administración”, 1ra ed. México, Thomson Learning, 2004, 835p.

[48] Nevado Cabello Ma. Victoria, “Introducción a las Bases de Datos Relacionales”, 1ra ed., Madrid España, Visión Libros, 2006, 117p.

[49] Powell T. A., “Diseño De Sitios Web”, 1ra ed., Madrid España, Mc Graw Hill, 2001.

[50] Patrick Cauldwell, “Servicios Web XML”, 1ra ed., Madrid España, Anaya Multimedia, 2002.

[51] M. Deitel Harvey y J. Deitel Paul, “Java cómo Programar”, 7ma ed. México, Person Education, 2010, 1268p.

[52] Gosling James, Joy Bill, Steeles Guy, Bracha Gilad, “The Java language: Specification” 2da ed., California EEUU, Sun Mycrosystems, 2000, 519p.

[53] Thayer H. Richard y Merlin Dorfman, “Software Requerements Engineering”, 2da ed., IEEE Computer Society Press, 1997.

[54] Matt Jones y Gary Marsden, “Mobile Interaction Design”, 1ra ed., John Wiley & Sons, 2006, 398p.

[55] Kenneth E. Kendall, Julie E. Kendall, “Análisis y diseño de sistemas”, 1ra ed., Pearson Education, 2005, 726 p.

[56] Banker, R., R. Kauffman and R. Kumar, “An Empirical Test of Object-Based Output Measurement Metrics in a Computer Aided Software Engineering (CASE) Environment”, Journal of Management Information Systems, 1994.

[57] Fowler Martin, Scott Kendall, “UML gota a gota”, 1ra ed., Pearson Education, 1999, 203 p.

[58] K.J. Van Der Poel, S.R. Schach. “A Software Metric for Cost Estimation and Efficiency Measurement in Data Processing System Development”, 1ra ed., J. System and Software, 1983, pp. 187-191.

[59] Qing Wang, Dietmar Pfahl, David M. Raffo, Paul Wernick, Eds, “Software Process Change: International Software Process Workshop and International Workshop on Software Process Simulation and Modeling, SPW/ProSim 2006, Shanghai, China, May 20-21, 2006, Proceedings”, 1ra ed. Springer, 2006, 356 p.

[60] Fowler Martin, “UML Distilled: a brief guide to the standard object modeling language”, 3rd ed., Addison – Wesley, 2004, 175p.

[61] Kenneth E. Kendall, Julie E. Kendall, “Análisis y diseño de sistemas”, 6ra ed., Prentice Hall, 2010, 726 p.

[62] Coronel Carlos, Morris Steven, Rob Peter, “Bases de Datos, Diseño, Implementación y Administración”, 1ra ed., Cengage Learning Editores, 2011, 255 p.

[63] Kroenke David M., “Procesamiento de bases de datos: fundamentos, diseño e implementación”, 8va ed., Pearson Educación, 2003, 671 p.

[64] Cuello Javier, Vittone José, “Diseñando apps para móviles”, 1va ed., José Vittone — Javier Cuello, 2013, 300 p.

[65] Adams E., “Optimizing Preventive Service of Software Products”. No 1, Vol. 28, IBM Research Journal, 1984, pp. 2-14.

[66] Elaine J. Y Vocolos F. I., “Experience with performance testing of software systems: Issues, approach and case study. IEEE transactions on software engineering”, Vol. 26, No. 12, 2000, pp. 1147-1156.

[68] Gibbs W. W., “Software’s Chronic Crisis. Scientific American”, No. 218, 1994, pp. 72-81.

[69] Littlewood B. Y Strigini L., “The Risks of Software. Scientific American”, No.1, Vol. 268, 1993, pp. 62-75.

[70] Voas J, “Software quality’s eight greatest myths”, IEEE Software, No. 5, Vol. 16, 1999, pp. 118 -120.

# Anexos

## Anexo 1 - Dominios De Aplicación De Software

**Software de Sistemas**: es un conjunto de programas escritos con la finalidad de proporcionar servicios a otros programas, sus características principales son interaccionar con el hardware de la computadora, operaciones concurrentes, recursos compartidos, administración de procesos, estructuras complejas de datos, por ejemplo compiladores, editores, herramientas para administración de archivos, sistemas operativos, manejadores, software de redes y procesadores de telecomunicaciones.

**Software de Aplicación**: son programas que resuelven una necesidad específica de negocios, procesan datos comerciales o técnicos de forma que se faciliten las operaciones de negocios y toma de decisiones, todo esto en tiempo real.

**Software de Ingeniería o Ciencia**: se caracterizan por el uso de algoritmos numéricos, su aplicación se refleja en astronomía, vulcanología, biología, automotriz, entre otras.

**Software Incrustado**: se aloja dentro de un producto o sistema, se usa para implementar y controlar funciones limitadas para el usuario final y para sistemas en sí, por ejemplo control del tablero de un horno de microondas.

**Software de Line a de Productos**: son diseñados para proporcionar una capacidad específica, se centran en mercados limitados y masivos de consumidores como control inventario, procesadores de texto, hojas de cálculo, entretenimiento, por mencionar algunos.

**Aplicaciones Web**: denominadas “webapps” este software se centra en redes y abarca una amplia gama de aplicaciones, desde el surgimiento de la web 2.0 sea logrado la integración con base de datos, además con la aparición de HTML5 y CSS3 se pueden crear aplicaciones nativas para celulares.

**Software de Inteligencia Artificial**: se caracteriza por no emplear algoritmos numéricos para resolver problemas complejos, las áreas de aplicación incluyen robótica, sistemas expertos, reconocimiento de patrones (videos e imágenes), redes neuronales, algoritmos genéticos, entre otras.

## Anexo 2 - Actividades De La Ingeniería Del Software

**Comunicación**: está etapa se enfoca en comunicarse y colaborar con el cliente, para obtener los requisitos que ayuden a definir las características y funciones del software.

**Planeación**: define el trabajo de ingeniería de software al describir la tareas técnicas por realizar, los riesgos probables, recursos que se requieren, los recursos del trabajo que se obtendrán y una programación de las actividades.

**Modelado**: es crear modelos con más y más detalles con el fin de entender mejor los requerimientos del software y el diseño que los satisfará.

**Construcción**: esta actividad se realiza la codificación y las pruebas que se requieren para descubrir errores en esté.

**Despliegue**: el software como una entidad completado como un incremento parcial terminado se entrega al consumidor, para la evaluación y retroalimentación para el proyecto.

## 

## Anexo 3 - Actividades Estructurales Del Proceso De Ingeniería De Software

**Seguimiento y control del proyecto de software**: permite que el equipo de software evalué el proceso de software y comparar con el plan del proyecto.

**Administración de riesgos**: esta actividad de evaluar los riesgos que pueden afectar al proyecto y la calidad del mismo.

**Aseguramiento de la calidad del software**: define cada una de las actividades que se ejecutan para garantizar la calidad del software.

**Revisiones técnicas**: esta actividad se encarga de activar los productos resultantes de la ingeniería de software antes que se propaguen a la siguiente actividad.

**Medición**: se encarga de definir y reunir mediciones de los procesos, proyectos y productos para apoyar al equipo a entregar el software.

**Administración de la configuración del software**: administra los efectos del cambio a lo largo del proceso del software.

**Administración de la reutilización**: se establecen criterios para generar componentes reutilizables productos de los componentes del software.

**Preparación y producción del producto del trabajo**: se encarga de agrupas las actividades requeridas para generar modelos, documentos, requisitos, formatos y listas.

## Anexo 4 - Procesos Del Ciclo De Vida Del Software

Procesos Principales del Ciclo de Vida

Los procesos principales del ciclo de vida son cinco, que dan servicio a las partes principales durante el ciclo de vida del software. Los procesos principales son (Loïc & Segovia, 2005):

1. Proceso de adquisición: Define las actividades de la organización que adquiere un sistema, producto software o servicio software.
2. Proceso de suministro: Define las actividades del proveedor de sistema, producto software o servicio software al adquiriente.
3. Proceso de desarrollo: Define las actividades del desarrollador del producto software.
4. Proceso de operación: Define las actividades del operador.
5. Proceso de mantenimiento: Define las actividades del responsable de mantenimiento, además migración y retirada del producto software.

Procesos de Apoyo del Ciclo de Vida

Un proceso de apoyo es el que apoya a otro proceso como parte esencial del mismo, con un propósito bien definido que contribuye al éxito y a la calidad del proyecto software. Los procesos de apoyo son (Loïc & Segovia, 2005).

1. Proceso de documentación: Define las actividades para el registro de la información producida por un proceso del ciclo de vida.
2. Proceso de gestión de la configuración: Define las actividades de la gestión de la configuración.
3. Proceso de aseguramiento de la calidad: Define las actividades para asegurar que los productos software y los procesos son conformes a sus requerimientos especificados y se ajustan a sus planes establecidos.
4. Proceso de verificación: Define las actividades para verificar los productos software del proyecto de software.
5. Proceso de validación: Define las actividades para validar los productos software del proyecto software.
6. Proceso de revisión conjunta: Define las actividades para evaluar el estado y productos de una actividad.
7. Proceso de auditoría: Define las actividades para determinar la conformidad con los requerimientos, planes y contrato.
8. Proceso de solución de problemas: Define las actividades para analizar y eliminar los problemas durante la ejecución del proceso de desarrollo, operación, mantenimiento u otros procesos.

Procesos Organizativos del Ciclo de Vida

Los procesos organizativos se emplean por una organización para establecer e implementar una infraestructura constituida por procesos y personal asociado al ciclo de vida y para mejorar continuamente esta infraestructura. Se usan habitualmente fuera del ámbito de proyectos y contratos específicos; sin embargo, la experiencia adquirida mediante dichos proyectos y contratos contribuye a la mejora de la organización. Los procesos organizativos son (Tuya & Román, 2007; Campderrich, 2003):

1. Proceso de gestión: Define las actividades básicas de gestión, incluyendo la gestión de proyectos, durante un proceso del ciclo de vida.
2. Proceso de infraestructura: Define las actividades básicas para establecer la infraestructura de un proceso del ciclo de vida.
3. Proceso de mejora de proceso: Define las actividades que una organización lleva a cabo para establecer, medir, controlar y mejorar sus procesos del ciclo de vida.
4. Proceso de recursos humanos: Define las actividades para conseguir personal adecuadamente capacitado.

## Anexo 5 - Tareas De La Ingeniería De Requerimientos

En base a lo dicho por Cri (1992) la ingeniería de requerimientos incluye siete tareas diferentes concepción, indagación, elaboración, negociación, especificación, validación y administración, algunas de estas tareas ocurren de manera paralela y todas se adaptan a la necesidad del proyecto.

**Concepción**: surge cuando se identifica una necesidad del negocio o se descubre un nuevo mercado o servicio potencial, en esta tarea se establece el entendimiento básico del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, así como la comunicación y colaboración de los participantes y el equipo de software.

**Indagación**

Esta tarea se enfoca en saber cuál es el objetivo u objetivos por el cual es creado el sistema o producto, que es lo que va a lograrse, como se ajusta el sistema o producto a las necesidades del negocio o problema y finalmente se va a usar el producto o sistema en operaciones o de manera cotidiana. Inmersos en esta tarea se encuentran tres tipos de problema:

1. De alcance: las fronteras del sistema están mal definidas, al especificar detalles técnicos innecesarios.
2. De entendimiento: los clientes o usuarios no saben completamente lo que necesitan y no entienden todo el dominio del problema, transmitiendo ideas erróneas o con falta de información, especifican requerimientos que entran en conflicto con las necesidades del cliente o de los usuarios.
3. De volatilidad: los requerimientos cambian con el tiempo.

**Elaboración**: es una tarea de expansión y refinamiento de la información, desarrollando modelos de los requerimientos, identificando diferentes aspectos de la función, comportamiento e información del software.

**Negociación**: se encarga de realizar negociaciones entre las partes afectadas por requerimientos que se encuentran en conflicto, ordenando los requerimientos según el grado de prioridad, logrando así realizar una evaluación de costos y riesgos, al concluir el proceso de negociación concluye con la eliminación, combinación y modificación de los requerimientos en conflictos.

**Especificación**: es un documento que se crea cuando se debe especificar una descripción detallada de cada uno de los aspectos del software antes que el proyecto comiese. Su costo es alrededor del 10 – 15 % del costo total del proyecto, también se toma en cuenta que un error de requisitos puede ser hasta 100 veces más costoso que un error de código.

**Validación**: analiza la especificación a fin de garantizar que los requerimientos han sido expuestos sin confusión. El equipo de revisión conformado por ingenieros de software, clientes, usuarios y otros participantes, encargados de analizar las especificaciones en búsqueda de errores de contenido, de interpretación, falta de información, inconsistencia.

**Administración de requerimientos**: es el conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y dar seguimiento a los requerimientos, asimismo si existe un cambio cualquier momento del proyecto.

## Anexo 6 - IEEE 830 Requisitos De Especificaciones De Software

**Alcance**: Esta es una práctica para la estructura de las especificaciones de requisitos de software. Describe el contenido y cualidades para un buen SRS y presenta varios ejemplos de SRS está dirigida para el desarrollo de especificaciones de requisitos de software.

Describe el proceso de creación de un producto y su contenido, se puede utilizar como modelo para un nivel más específico del SRS.

**Definiciones**

* Contrato: documento legal aprobado por el cliente y el proveedor, incluye requisitos técnicos y organizacionales, costos y la calendarización del producto.
* Cliente: persona o personas que pagan por el producto y comúnmente deciden los requisitos.
* Proveedor: es la entidad que proveerá el producto al consumidor, para este caso el consumidor o cliente será el mismo proveedor.
* Usuario: persona o personas que operan o interactúan directamente con el producto.

**Naturaleza del SRS**

Un SRS es una especificación para un software, programa o conjunto de programas que realizan ciertas funciones en un entorno específico. A continuación detallaremos las siguientes características en un SRS: funcionalidad; interfaces externas, rendimiento, atributos, limitaciones de diseño e implementación.

**Entorno de un SRS**

Los SRS tienen un rol específico dentro del proceso específico dentro del proceso de desarrollo del software, la escritura del mismo debe tener cuidado en no ir más allá de los límites de este rol, lo que significa que:

* Debe definir correctamente todos los requerimientos del software.
* No debe describir los detalles del diseño o implementación.
* No debe describir restricciones adicionales sobre el software.

**Características de un SRS**

**Correcto**: un SRS es correcto si y solo si cada uno de los requerimientos declarados en este documento son satisfechos por el software.

**No ambiguo**: cada uno de los requerimientos tiene una sola interpretación. Requiere que cada característica del producto sea descripta con un solo término.

* Trampas de lenguaje.
* Requerimientos de especificaciones de idioma.
* Herramientas de representación.
* Enfoque orientado a objetos.
* Enfoque basado en procesos.
* Enfoque en comportamiento.

**Complejo**: un SRS es complejo si y solo si este incluye los siguientes elementos:

1. Todo requerimiento significante como funcionalidad, atributos, performance, restricciones de diseño e interfaces externas.
2. Definición de la respuesta del software a todas las clases de entradas realizables en todas las situaciones posibles, tanto para entradas validas como invalidas.
3. Etiquetas y referencias de todas las figuras, tablas y diagramas en el SRS y definición de todos los términos y unidades de medida.

**Consistencia**

**Consistencia interna**: un SRS es coherente si y solo si ningún conjunto de los requerimientos individuales entra en conflicto. Dentro de los conflictos surgidos se encuentra las siguientes situaciones:

1. Las características específicas de un objeto del mundo real pueden entrar en conflicto.
2. Conflictos de tipo lógico o temporal entre acciones específicas.
3. Dos o más requerimiento s pueden describir el mismo objeto pero usando diferentes términos.

**Clasificación por importación y estabilidad**: se está ordenado por importancia y estabilidad si cada requerimiento tiene un identificador que indique estas características.

* **Grado de estabilidad**: Este grado se expresa en número de cambios que se esperan a cualquier requerimiento en base a la experiencia o conocimientos en los eventos que afectan a la organización, funciones y personas.
* **Grado de necesidad**
* **Esenciales**: el software no se aceptara si no cumple de manera adecuada con los requerimientos.
* **Condicional**: Son requerimientos que mejorarían al software pero no se requieren para que este sea aceptado.
* **Opcionales**: Cosas extra que solo hacerlas si te sobra tiempo.

**Verificabilidad**: es verificable si cada uno de sus requerimientos lo son, los cuales lo son si existe un proceso finito que lo reconozca. Si un método no puede ser ideado para determinar si el software cumple un requerimiento particular, entonces el requerimiento debería ser removido o redefinido para que lo sea.

**Modificabilidad**: es modificable si su estructura y estilo son tales que cualquier cambio en los requerimientos se pueden realizar fácilmente, completamente y consistentemente mientras se conserve la estructura y estilo. Está condicionada a:

* Tener una organización coherente y fácil de usar con tablas de contenido, índices y explicitas referencias cruzadas.
* No ser redundante.
* Expresar cada requerimiento por separado, antes de mezclarlos con otros.

**Rastreabilidad**: es detectable si el origen de cada uno de los requerimientos es claro y facilita referenciar a cada requerimiento en el futuro desarrollo del documento.

## Anexo 7 - El Modelo Constructivo De Costos O COCOMO II

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo de software. Está asociado a los ciclos de vida modernos (Qing Wang, et al., 2006). Al igual que el modelo COCOMO 81 comentado en el punto anterior, COCOMO II está dividido en otros modelos o niveles distintos:

* El modelo de Composición de Aplicaciones: Indicado para la estimación del esfuerzo de proyectos construidos con herramientas modernas de construcción de interfaces gráficos para usuario. Usa puntos de aplicación como una entrada ().
* El modelo de Diseño anticipado o temprano: Este modelo puede utilizarse para calcular el esfuerzo inicial basado en puntos objeto o de función y en una opción de diseño. Utiliza un. Está basado en Punto de Función sin ajustar o KSLOC (Miles de Líneas de Código Fuente).
* El modelo de Reutilización: se estima el esfuerzo de integración de componentes reutilizables generados automáticamente y utiliza las líneas de código generadas como entradas.
* El modelo Post-Arquitectura: Es el modelo más detallado, se utiliza una vez que drivers de coste, nuevas reglas para el recuento de líneas y nuevas ecuaciones. Utiliza el de líneas de código fuente como entradas.

La tabla 25 hace referencia a la tabla del factor de escala en el proyecto. El factor de precedente (PREC) toma el grado de experiencia previa del personal implicado en el proyecto a desarrollar, tanto en el aspectos organizacional, conocimiento de hardware y software. El factor de flexibilidad (FLEX) considera el nivel de experiencia en el cumplimiento de los requerimientos preestablecidos, plazos de tiempo y especificaciones de la interfaz. Para analizar ambos factores se utiliza la tabla 25 del modelo COCOMO.

Tabla – Factores de escala para analizar PREC y FREX

|  |  |  |  |
| --- | --- | --- | --- |
| Características | Muy Bajo | Nominal | Extra Alto |
| **Bajo** | **Alto** | **Muy Alto** |
| Precedencia | | | |
| Entendimiento organizacional de los objetivos del producto. | General | Considerable | Total |
| Experiencia en el trabajo con software relacionado. | Moderada | Considerable | Amplia |
| Desarrollo concurrente de nuevo hardware y procedimientos operacionales. | Abundante | Moderado | Escaso |
| Necesidad de innovación en el procesamiento de datos, arquitectura y algoritmos. | Considerable | Alguno | Mínima |
| Flexibilidad de desarrollo | | | |
| Necesidad de conformar requerimiento pre-establecidos | Total | Considerable | Básica |
| Necesidad de conformar especificaciones externas de la interface | Total | Considerable | Básica |
| Estímulo por terminación temprana | Elevado | Medio | Bajo |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

El factor de arquitectura y determinación del riesgo (RESL) involucra aspectos relacionados al conocimiento de los ítems de riesgo crítico y al modo de abordarlos dentro del proyecto. Es el resultado promedio de acuerdo a la tabla que se presenta a continuación.

Tabla – Componentes para calcular el factor de escala RESL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Características | Muy Bajo | Bajo | Nominal | Alto | Muy Alto | Extra Alto |
| Planificación de la administración de riesgo, identificando todos los ítems de riesgo y estableciendo hitos de control para su solución por medio de la revisión del diseño del producto (PDR). | Ninguna | Pequeña | Algo | General | En gran medida | Completo |
| Cronograma, presupuesto e hitos internos especificados en el PDR, compatibles con el plan de administración de riesgo | Ninguna | Pequeña | Algo | General | En gran medida | Completo |
| Porcentaje del cronograma dedicado a la definición de la arquitectura de acuerdo a los objetivos generales del producto | 5 | 10 | 17 | 25 | 33 | 40 |
| Porcentaje de arquitecturas de software disponibles para el proyecto. | 20 | 40 | 60 | 80 | 100 | 120 |
| Herramientas disponibles para resolver ítems de riesgo, desarrollando y verificando las especulaciones de arquitecturas | Ninguna | Pocas | Algunas | Buenas | Muy Buenas | Completas |
| Nivel de incertidumbre en factores claves de la arquitectura: interface de usuario, hardware, tecnología, performance. | Extremo | Significativo | Considerable | Medio | Pocos | Muy Pocos |
| Cantidad y grado de criticidad de ítems de riesgo. | >10 Critico | 5 -10 Critico | 2 - 4 Critico | 1 Critico | > 5 No critico | < 5 No Critico |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

El factor de cohesión de equipo (TEAM) tiene en cuenta las dificultades de sincronización entre los particulares del proyecto: usuarios, clientes, desarrolladores, etc. El valor se calcula como promedio ponderado de las características de la siguiente tabla.

Tabla – Componentes del factor TEAM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Características | Muy Bajo | Bajo | Nominal | Alto | Muy Alto | Extra Alto |
| Compatibilidad entre los objetivos y culturas de los integrantes del equipo. | Poca | Alguna | Básica | Considerable | Fuerte | Total |
| Habilidad y predisposición para conciliar objetivos. | Poca | Alguna | Básica | Considerable | Fuerte | Total |
| Experiencia en el trabajo en equipo. | Ninguna | Poca | Poca | Básica | Considerable | Vasto |
| Visión compartida de objetivos y compromisos compartidos. | Ninguna | Poca | Poca | Básica | Considerable | Amplia |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

El factor de madurez del proceso o PMAT se basa en el modelo de capacidad madurez (CMM o Capability Maturity Model) propuesto por el Software Engineering Institute y existen dos formas de calcularlos.

* Se asignas valores correspondientes según el nivel de madurez de la organización como se presenta en la siguiente tabla.

Tabla – Factor PMAT según el nivel CMM

|  |  |
| --- | --- |
| Nivel de CMM | PMAT |
| 1 - Mitad Inferior | Muy Bajo |
| 1 - Mitad Inferior | Bajo |
| 2 | Normal |
| 3 | Alto |
| 4 | Muy Alto |
| 5 | Extra Alto |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

* Se basa en las dieciocho áreas de procesos claves (KPAs) del modelo SEI. Estableciendo porcentajes en cada una de las áreas, como se muestra a continuación. Después de cumplir los niveles de cumplimiento de cada KPA se calcula por la ecuación 11 del factor PMAT.

Tabla – Nivel de cumplimiento de objetivos KPA

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Áreas de Procesos Claves | Casi Siempre (90%) | A menudo (60-90%) | La mitad de las veces (40-60%) | A veces (10-40%) | Casi nunca (<10%) | No se aplica | No se conoce |
| Administración de Requerimientos |  |  |  |  |  |  |  |
| Planificación del Proyecto de Software |  |  |  |  |  |  |  |
| Seguimiento y supervisión del Proyecto de Software |  |  |  |  |  |  |  |
| Administración de Subcontratos |  |  |  |  |  |  |  |
| Aseguramiento de la Calidad |  |  |  |  |  |  |  |
| Administración de la Configuración |  |  |  |  |  |  |  |
| Objetivo del Proceso de Organización |  |  |  |  |  |  |  |
| Definición del Proceso de Organización |  |  |  |  |  |  |  |
| Programa de Entrenamiento Administración Integrada de Software |  |  |  |  |  |  |  |
| Ingeniería del Producto |  |  |  |  |  |  |  |
| Coordinación entre Grupos |  |  |  |  |  |  |  |
| Revisión por Pares |  |  |  |  |  |  |  |
| Administración Cuantitativa |  |  |  |  |  |  |  |
| Administración de la Calidad |  |  |  |  |  |  |  |
| Prevención de Defectos |  |  |  |  |  |  |  |
| Administración de las Tecnologías de Cambio |  |  |  |  |  |  |  |
| Administración de los Procesos de Cambio |  |  |  |  |  |  |  |

**Nota**: Boehm B.W.,Clark B., Horowizt E., Westland C., Madachy R., Selby R.,The COCOMO 2.0 Software Cost Estimation Model.

Ecuación – Determinar la madurez del proceso.

1. IBM (International Business Machines) es una empresa multinacional estadounidense de tecnología y consultoría. [↑](#footnote-ref-1)
2. SMS (Short Message Service) es un servicio mensajes de cortos disponible en los teléfonos móviles de la segunda generación de dipositivos. [↑](#footnote-ref-2)
3. Hz o hercio es la unidad de frecuencia de Sistema Internacional de Unidades. [↑](#footnote-ref-3)
4. PTT o pulsa y habla es un método para hablar en línea half-dúplex de comunicación, [↑](#footnote-ref-4)
5. NMT fue el primer sistema celular completamente automático y opero a partir del 1 de octubre de 1981. [↑](#footnote-ref-5)
6. T-Mobile es un operador de red móvil establecido en Bonn (Alemania). Es un subsidiario de Deutsche Telekom y pertenece a la alianza de negocio de FreeMove. La t representa "Telekom". [↑](#footnote-ref-6)
7. QSM es una compañía pionera en la estimación de software desde 1978. [↑](#footnote-ref-7)
8. Modelo de Madurez de Capacidades o CMM (Capability Maturity Model), es un modelo de evaluación de los procesos de una organización. [↑](#footnote-ref-8)