



UNIVERSIDAD TÉCNICA DE COTOPAXI

UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

COMPUTACIONALES

TITULO:

“ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

**TESIS PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIEROS EN
INFORMÁTICA Y SISTEMAS COMPUTACIONALES**

AUTORA:

Maigualca Moreno Maritza Elena

DIRECTOR DE TESIS:

Ing. Javier Montaluisa Yugla

ASESORA DE TESIS:


Dra. Anita Chancusi Heredia

Latacunga, Noviembre del 2012

DECLARACIÓN AUTORA DE TESIS

La presente tesis titulada: “ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI” fue realizada por la señorita Maritza Elena Maigualca Moreno, egresada de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas Carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi responsabilizándose de los criterios y opiniones emitidas en esta investigación.

Autora.




Maritza Maigualca
C.I. 050328822-7
EGRESADA U.T.C.

AVÁL DIRECTOR DE TESIS

En calidad de Director del Trabajo de Investigación titulado “ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI” propuesto por la postulante Maritza Elena Maigualca Moreno considero que dicho informe cumple con los requerimientos metodológicos y aportes científicos-técnicos suficientes para ser sometidos a evaluación del Tribunal de Validación de Tesis que el Honorable Consejo Académico de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe para su correspondiente estudio y calificación.

Atentamente,



Ing. Javier Montaluisa
DIRECTOR DE TESIS

AVÁL TRIBUNAL DE TESIS



UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE INGENIERÍA Y APLICADAS
CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

Latacunga-Ecuador

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de Miembros del Tribunal de Grado aprueban el presente informe técnico de investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Unidad Académica de Ciencias de la Ingeniería y Aplicadas; por cuanto, la postulante: Maigualca Moreno Maritza Elena con el título de tesis: "ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI", ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Defensa de Tesis.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, 31 de Octubre del 2012

Para constancia firman:


Ing. Patricio Navas
PRESIDENTE


Ing. Diana Marín
MIEMBRO


Ing. Galo Flores
OPOSITOR

CERTIFICADO RESUMEN DE TESIS



UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS ADMINISTRATIVAS
Y HUMANÍSTICAS

Latacunga - Ecuador

CERTIFICADO RESUMEN DE TESIS

Yo, Msc. Amparo Romero, en calidad de docente del Centro de Idiomas de la Universidad Técnica de Cotopaxi, tengo a bien **CERTIFICAR** haber revisado el resumen de tesis de la señorita Maigualca Moreno Maritza Elena con C.I. N°. 050328822-7, egresada de la Carrera de Ingeniería en Informática y Sistemas Computacionales con el Tema: “ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”, en el periodo académico Enero-Julio del 2012 el cual se encuentra bien estructurado por lo que doy fe del presente trabajo.

Por tal motivo faculto a la interesada hacer uso del presente certificado como a bien considere.

Atentamente,

Msc. Amparo Romero
C.I. 050136918-5

DOCENTE CENTRO DE IDIOMAS

AVÁL DE IMPLEMENTACIÓN



UNIVERSIDAD TÉCNICA DE COTOPAXI
DIRECCIÓN DE SERVICIOS INFORMÁTICOS
Latacunga – Ecuador

CERTIFICACIÓN

En calidad de Director de la Dirección de Servicios Informáticos (E) de la Universidad Técnica de Cotopaxi me permito CERTIFICAR que: una Aplicación Móvil se subió a la Página Web de la Unidad de Ciencias de la Ingeniería y Aplicadas de la carrera de Sistemas, por lo cual está disponible para su descarga desde la Página Web de nuestra Institución; Aplicación que es presentada por la Señorita MAIGUALCA MORENO MARITZA ELENA, portadora de la cédula de ciudadanía N° 050328822-7, Estudiante Egresada de la Especialidad de Ingeniería en Informática y Sistemas Computacionales.

Es todo cuanto puedo certificar, en honor a la verdad facultando al interesado hacer uso del presente documento como que estime conveniente.

Latacunga septiembre 28,2012

Atentamente

“POR LA VINCULACIÓN DE LA UNIVERSIDAD CON EL PUEBLO”


Ing. Adrián Mena Rojas
DIRECTOR DE SERVICIOS INFORMÁTICOS (E)

AGRADECIMIENTO

Al término de esta etapa de mi vida, quiero expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión me alentaron a lograr esta hermosa realidad. Primero a Dios por estar conmigo en cada paso que doy, por fortalecer mi mente y llenarla de sabiduría, por darme la vida y por permitirme escribir estas líneas.

Un sincero agradecimiento a las autoridades de la Universidad Técnica de Cotopaxi por abrirme las puertas desde el primer día que ingrese a sus aulas y a los docentes por brindarme cada uno de sus conocimientos y experiencias a la hora de educarme.

Un agradecimiento especial al Ing. Javier Montaluisa, DIRECTOR DE TESIS y la Dra. Anita Chancusi, ASESORA DE TESIS gracias por su tiempo y paciencia incondicional a la hora de enseñar porque juntos sin egoísmo, guiaron el desarrollo de este trabajo desde el inicio hasta lograr la meta final.

Gracias a todos quienes de una u otra manera me apoyaron para culminar este trabajo.

Maritza Elena

DEDICATORIA

A la Virgen mi madre del cielo por guiarme y cuidarme siempre aun en los momentos difíciles ella no me abandonó, gracias a su apoyo espiritual hoy le dedico estas palabras con gran humildad.

Un Dios le pague por concederme mi mejor regalo mi amada familia ya que gracias a su esfuerzo y apoyo incondicional que me brindaron durante el transcurso de mi vida y mis estudios hoy a llegado la hora de dedicarles la mejor herencia recibida mi anhelada profesión.

A mis padres Mario y Consuelo quienes sin limitar esfuerzo alguno sacrificaron gran parte de su vida para educarme hasta el dia de hoy, brindándome sus mejores consejos, apoyo moral, económico y sobre todo sus inolvidables bendiciones.

A mis hermanos Rossy y Patricio a quienes jamás encontraré la forma de agradecer el cariño y comprensión brindado en los momentos buenos y malos de mi vida por apoyarme y darme un aliento a seguir adelante.

Maritza Elena

INDICE DE CONTENIDOS

CONTENIDOS	PÁGS.
PORTADA.....	i
DECLARACIÓN AUTORÍA DE TESIS.....	ii
AVÁL DIRECTOR DE TESIS.....	iii
AVÁL TRIBUNAL DE TESIS.....	iv
CERTIFICADO RESUMEN DE TESIS.....	v
AVÁL DE IMPLEMENTACIÓN.....	vi
AGRADECIMIENTO.....	vii
DEDICATORIA.....	viii
INDICE DE CONTENIDOS.....	ix
INDICE DE GRAFICOS.....	xiv
INDICE DE DIAGRAMAS.....	xvi
INDICE DE ANEXOS.....	xvi
INDICE DE TABLAS.....	xvii
RESUMEN.....	xviii
SUMMARY.....	xix
INTRODUCCIÓN.....	xx
CAPÍTULO I.....	24
FUNDAMENTACIÓN TEÓRICA.....	24
1.1. <i>Introducción</i>	24
1.2. <i>Tecnologías de Comunicación Inalambrica</i>	25
1.2.1. <i>Historia</i>	25
1.2.2. <i>Clasificación de las Tecnologías de Comunicación Inalámbrica</i>	26
1.3. <i>Telefonía celular</i>	27
1.3.1. <i>Breve historia</i>	28
1.3.2. <i>Ventajas</i>	29
1.3.3. <i>Generaciones de la Telefonía Móvil</i>	30
1.3.3.1. <i>La primera generación 1G</i>	30
1.3.3.2. <i>La segunda generación 2G</i>	30

1.3.3.3. La tercera generación 3G	30
1.3.3.4. La cuarta generación 4G	31
1.4. <i>Dispositivos Móviles</i>	31
1.5. <i>Software Libre</i>	32
1.5.1. Licencias GPL	33
1.5.2. Bondades de software libre	33
1.6. <i>Sistema Operativo</i>	34
1.6.1. Tipos de Sistemas Operativos para dispositivos móviles	35
1.6.1.1. iOS	35
1.6.1.2. Android	35
1.6.1.3. RIM OS	36
1.6.1.4. Symbian OS	36
1.6.1.5. Windows Phone	36
1.7. <i>Análisis de Herramientas</i>	37
1.8. <i>Estudio de Herramientas libres</i>	39
1.8.1. Lenguajes de Programación	39
1.8.1.1. Lenguaje Java	39
1.8.1.1.1. Características	40
1.8.1.2. Lenguaje XML	41
1.8.1.2.1. Ventajas	42
1.8.2. Programacion Orientada a Objetos	42
1.8.3. JVM	43
1.8.4. JRE	44
1.8.5. JDK	44
1.8.5.1. Estructura JDK	45
1.8.6. Android SDK	46
1.8.6.1. Maquina Virtual Dalvik	46
1.8.6.2. Emulador Android AVD/ Dalvik	47
1.8.6.3. ADT	48

1.8.6.4. Base de Datos.....	48
1.8.6.4.1. SQLite	48
1.8.7. Ide's de Programación.....	49
1.8.7.1. Eclipse	49
1.8.8. Sistema operativo Android.....	50
1.8.8.1. Historia	50
1.8.8.2. Definiciones.....	51
1.8.8.3. Características de Android.....	52
1.8.8.4. Arquitectura de Android	52
1.9. Metodología de Desarrollo.....	54
1.9.1. Metodologías ágiles.....	55
1.9.2. Metodologías ágiles para el desarrollo de software móvil.....	56
1.9.2.1. Características y requerimientos del entorno móvil.....	58
CAPÍTULO II	60
DESCRIPCIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	60
2.1. Introducción.....	60
2.2. Caracterización de la Universidad Técnica de Cotopaxi.....	61
2.2.1. Reseña Histórica de la Universidad Técnica de Cotopaxi.....	61
2.2.2. Misión Institucional	62
2.2.3. Visión	62
2.3. Organigrama estructural	63
2.4. Descripción de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas ..	64
2.4.1. Misión	64
2.4.2. Visión	64
2.5. Descripción de la Carrera de Ingeniería en Informática y Sistemas Computacionales	64
2.5.1. Misión	65
2.5.2. Visión	66
2.5.3. Perfil Profesional	66

2.6. <i>Unidad de Estudio</i>	68
2.6.1. <i>Población</i>	68
2.6.2. <i>Muestra</i>	68
2.7. <i>Análisis e interpretación de resultados de las encuestas realizadas a docentes y alumnos de la Carrera de Sistemas</i>	69
2.8. <i>Verificación de la hipótesis</i>	79
2.9. <i>Metodología para el desarrollo de aplicaciones móviles</i>	80
2.9.1. <i>Metodología XP (Programación Extrema)</i>	80
2.9.1.1. <i>Descripción de Extreme programming</i>	80
2.9.1.2. <i>Características de Xp</i>	81
2.9.1.3. <i>Fases de la Metodología Xp</i>	82
2.9.1.3.1. <i>Fase de Exploración</i>	83
2.9.1.3.2. <i>Fase de Planificación</i>	83
2.9.1.3.3. <i>Fase de Iteraciones</i>	83
2.9.1.3.4. <i>Fase Puesta en Producción</i>	84
2.9.1.3.5. <i>Fase de Mantenimiento</i>	84
2.9.1.3.6. <i>Fase Muerte del Proyecto</i>	84
CAPITULO III	85
DESARROLLO DE LA PROPUESTA	85
3.1. <i>Tema</i>	85
3.2. <i>Presentación</i>	85
3.3. <i>Justificación</i>	86
3.4. <i>Objetivos</i>	87
3.5. <i>Factibilidad de aplicar la propuesta</i>	87
3.6. <i>Desarrollo de la aplicación</i>	88
3.6.1. <i>Fase de Exploración</i>	88
3.6.1.1. <i>JDK(Java Development Kit)</i>	88
3.6.1.2. <i>IDE ECLIPSE</i>	92
3.6.1.3. <i>SDK</i>	94
3.6.1.4. <i>BASE DE DATOS SQLite</i>	100

3.6.1.5. DATAFRAMEWORK.....	102
3.6.2. Fase de Planificación	102
3.6.2.1. Ámbito	103
3.6.2.2. Obtener requisitos	103
3.6.2.2.1. Identificación de los interesados	103
3.6.2.2.2. Identificación de Actores	103
3.6.2.2.3. Requerimientos Informales	104
3.6.2.2.4. Historias de Usuarios o especificación de requisitos (Requerimientos)	106
3.6.2.3. Análisis de Información	111
3.6.2.3.1. Priorización y estimación	111
3.6.2.3.2. Distribución Funcional	111
3.6.3. Fase de Iteraciones.....	112
3.6.3.1. Diseño	113
3.6.3.1.1. Diagramas de casos de uso	113
3.6.3.1.2. Diagrama de actividad	119
3.6.3.1.3. Diseño de datos	122
3.6.3.1.4. Mapa de la aplicación	126
3.6.3.2. Construcción	126
3.6.3.2.1. Diseño de la experiencia de usuarios	126
3.6.3.2.2. Prototipo de interfaces	127
3.6.3.2.3. Codificación	133
3.6.3.3. Pruebas	170
3.6.4. Fase de Puesta en producción - Implementación	176
3.6.5. Fase de Mantenimiento	176
3.6.6. Fase Muerte del proyecto	176
3.7. <i>Conclusiones</i>	177
3.8. <i>Recomendaciones</i>	178
3.9. <i>Glosario de Términos</i>	179
3.10. <i>Referencias Bibliográficas</i>	184
3.11. <i>Anexos</i>	186

INDICE DE GRAFICOS

CONTENIDOS

PÁGS.

CAPITULO I

GRAFICO N° 1. 1.	Telefonía celular.....	27
GRAFICO N° 1. 2.	Ejemplificación del leguaje de marcas extendido XML.....	40
GRAFICO N° 1. 3.	Proceso de generación y ejecución de obtención de resultados.....	43
GRAFICO N° 1. 4.	Estructura de directoris Jdk.....	44
GRAFICO N° 1. 5.	Codificación eficiente basada en procesadores pequeños para dispositivos.....	46
GRAFICO N° 1. 6.	Arquitectura del Sistema Operativo Android.....	53

CAPITULO II

GRAFICO N° 2. 1.	Organigrama Estructural U.T.C.....	62
GRAFICO N° 2.2.	PREGUNTA N°1.....	68
GRAFICO N° 2.3.	PREGUNTA N°2.....	69
GRAFICO N° 2.4.	PREGUNTA N°3.....	70
GRAFICO N° 2.5.	PREGUNTA N°4.....	71
GRAFICO N° 2.6.	PREGUNTA N°5.....	72
GRAFICO N° 2.7.	PREGUNTA N°6.....	73
GRAFICO N° 2.8.	PREGUNTA N°7.....	74
GRAFICO N° 2.9.	PREGUNTA N°8.....	75
GRAFICO N° 2.10.	PREGUNTA N°9.....	76
GRAFICO N° 2.11.	PREGUNTA N°10.....	77
GRAFICO N° 2.12.	Desarrollo de un Sistema usando XP.....	80
GRAFICO N° 2.13.	Bases de eXtreme Programming.....	81
GRAFICO N° 2.14.	Proceso de eXtreme Programming.....	81

CAPITULO III

GRAFICO N° 3.1.	Ejecutar Archivo JDK.....	88
GRAFICO N° 3.2.	Instalación JDK.....	88
GRAFICO N° 3.3.	Configuración variable de entorno.....	89
GRAFICO N° 3.4.	Configuración variable path.....	89
GRAFICO N° 3.5.	Configuración variable Classpath.....	90
GRAFICO N° 3.6.	Consola DOS para actualizar interprete de Java.....	90
GRAFICO N° 3.7.	Archivo zip de eclipse descomprimido.....	91
GRAFICO N° 3.8.	Ejecutar archivo eclipse.exe.....	91
GRAFICO N° 3.9.	Pantalla bienvenida de eclipse.....	92
GRAFICO N° 3.10.	Archivo zip de sdk descomprimido.....	93
GRAFICO N° 3.11.	Instalacion paquetes necesarios del sdk manager.exe.....	94

GRAFICO N° 3.12.	Configuración plugin de android para eclipse.....	94
GRAFICO N° 3.13.	Lista de herramientas a descargarse para el plugin.....	95
GRAFICO N° 3.14.	Actualizar el adt android en eclipse.....	96
GRAFICO N° 3.15.	Instalacion targets de SDK.....	96
GRAFICO N° 3.16.	Ejecutar e instalar el AVD manager.exe.....	97
GRAFICO N° 3.17.	Creando un nuevo dispositivo virtual AVD.....	97
GRAFICO N° 3.18.	Configurando el AVD.....	98
GRAFICO N° 3.19.	Inicializando el AVD.....	98
GRAFICO N° 3.20.	Pantalla inicial del emulador.....	99
GRAFICO N° 3.21.	Descargando el sqlite manager para Firefox.....	99
GRAFICO N° 3.22.	Abrir base de datos sqlite.....	100
GRAFICO N° 3.23.	Área de trabajo de sqlite.....	100
GRAFICO N° 3.24.	Elemento Actor de Caso de Uso.....	112
GRAFICO N° 3.25.	Elemento caso de uso.....	112
GRAFICO N° 3.26.	Elementos de un Diagrama de Actividad.....	118
GRAFICO N° 3.27.	Pantalla Inicial.....	126
GRAFICO N° 3.28.	Pantalla Menú Principal.....	126
GRAFICO N° 3.29.	Pantalla Jugador (consultar jugador).....	127
GRAFICO N° 3.30.	Pantalla Jugador (insertar jugador).....	127
GRAFICO N° 3.31.	Pantalla Preguntas (ingreso de preguntas y respuesta).....	128
GRAFICO N° 3.32.	Pantalla Preguntas (Ingreso de opciones a cada pregunta).....	129
GRAFICO N° 3.33.	Pantalla jugar.....	129
GRAFICO N° 3.34.	Pantalla Resumen (resumen lista jugadores).....	130
GRAFICO N° 3.35.	Pantalla Resumen (resumen lista partida).....	130
GRAFICO N° 3.36.	Pantalla Resumen (resumen de juego).....	131
GRAFICO N° 3.37.	Pantalla Acerca de (datos de autoria de aplicacion).....	131
GRAFICO N° 3.38.	Pantalla Instrucciones (instrcciones a seguir para iniciar juego).....	132
GRAFICO N° 3.39.	Estructura proyecto Android.....	132
GRAFICO N° 3.40.	La carpeta SRC (HolamundoActivity.java).....	133
GRAFICO N° 3.41.	La carpeta GEN	134
GRAFICO N° 3.42.	La carpeta BIN	135
GRAFICO N° 3.43.	La carpeta RES	136
GRAFICO N° 3.44.	Interfaz grafica de proyecto.....	137
GRAFICO N° 3.45.	Layout (main.xml).....	139
GRAFICO N° 3.46.	Layout (menú_principal.xml).....	140
GRAFICO N° 3.47.	Layout (jugador_consultar.xml).....	142
GRAFICO N° 3.48.	Layout (jugador_modificar.xml).....	144
GRAFICO N° 3.49.	Layout (jugador_insertar.xml).....	146
GRAFICO N° 3.50.	Layout (preguntas.xml).....	147
GRAFICO N° 3.51.	Layout (preguntas_listado.xml).....	150
GRAFICO N° 3.52.	Layout(opciones_pregunta.xml).....	152

GRAFICO N° 3.53.	Layout(eliminar opcion).....	155
GRAFICO N° 3.54.	Layout (jugar_layout2).....	155
GRAFICO N° 3.55.	Layout (resumen_listado_jugadores.xml).....	161
GRAFICO N° 3.56.	Layout (resumen_listado_partidas.xml).....	163
GRAFICO N° 3.57.	Layout (resumen_resumen.xml).....	165
GRAFICO N° 3.58.	Layout (acerca_de.xml).....	167
GRAFICO N° 3.59.	Layout(instrucciones).....	168

INDICE DE DIAGRAMAS

CONTENIDOS	PÁGS.	
CAPITULO III		
DIAGRAMA N° 3.1.	Diagrama de casos de uso aplicación mida su inteligencia.....	113
DIAGRAMA N° 3.2.	Diagrama de casos de uso gestión jugador.....	114
DIAGRAMA N° 3.3.	Diagrama de casos de uso gestión preguntas.....	115
DIAGRAMA N° 3.4.	Diagrama de casos de uso gestión jugar.....	116
DIAGRAMA N° 3.5.	Diagrama de casos de uso gestión resumen partida.....	117
DIAGRAMA N° 3.6.	Diagrama de actividad 1 (JUGADOR).....	119
DIAGRAMA N° 3.7.	Diagrama de actividad 2 (ADMINISTRADOR).....	120
DIAGRAMA N° 3.8.	Diagrama modelo conceptual de la base de datos.....	121
DIAGRAMA N° 3.9.	Diagrama modelo físico de la base de datos.....	122

INDICE DE ANEXOS

CONTENIDOS	PÁGS.	
CAPITULO III		
ANEXO N° 3.1.	Modelo de Encuesta aplicada a Docentes y Estudiantes de la Carrera de Sistemas.....	186
ANEXO N° 3.2.	Celulares con Android.....	188
ANEXO N° 3.3.	Lenguajes de Programación.....	189
ANEXO N° 3.4.	IDES de Programación.....	190
ANEXO N° 3.5.	Bases de Datos Embebidas.....	191
ANEXO N° 3.6.	Dispositivos Virtuales para Android.....	193

INDICE DE TABLAS

CONTENIDOS	PÁGS.
CAPITULO I	
TABLA N° 1. 1. Bases para métodos ágiles y clásicos.....	55
TABLA N° 1. 2. Características básicas ágiles y rasgos en el desarrollo de software móvil.....	56
CAPITULO II	
TABLA N° 2. 1. Población a investigar.....	67
TABLA N° 2.2. PREGUNTA N°1.....	68
TABLA N° 2.3. PREGUNTA N°2.....	69
TABLA N° 2.4. PREGUNTA N°3.....	70
TABLA N° 2.5. PREGUNTA N°4.....	71
TABLA N° 2.6. PREGUNTA N°5.....	72
TABLA N° 2.7. PREGUNTA N°6.....	73
TABLA N° 2.8. PREGUNTA N°7.....	74
TABLA N° 2.9. PREGUNTA N°8.....	75
TABLA N° 2.10. PREGUNTA N°9.....	76
TABLA N° 2.11. PREGUNTA N°10.....	77
CAPITULO III	
TABLA N° 3.1. Actores de la aplicación.....	102
TABLA N° 3.2. Primera historia de usuario(Introduccion).....	105
TABLA N° 3.3. Segunda historia de usuario (Gestion Menu de Opciones).....	105
TABLA N° 3.4. Tercera historia de usuario(Gestion Jugadores).....	106
TABLA N° 3.5. Cuarta historia de usuario(Gestion Preguntas).....	106
TABLA N° 3.6. Quinta historia de usuario(Gestion Jugar).....	107
TABLA N° 3.7. Sexta historia de usuario(Gestion jugar II).....	107
TABLA N° 3.8. Séptima historia de usuario(Gestion Jugar III).....	108
TABLA N° 3.9. Octava historia de usuario(Gestion Resumen).....	108
TABLA N° 3.10. Novena historia de usuario(Gestion Acerca de).....	109
TABLA N° 3.11. Decima historia de usuario(Gestion Instrucciones).....	109
TABLA N° 3.12. Priorizacion de Historias de Usuario.....	110
TABLA N° 3.13. Módulos de aplicación (Distribución Funcional).....	111
TABLA N° 3.14. Plan de Verificación del Requerimientos con el Sistema.....	171
TABLA N° 3.15. Pruebas Funcionales. Administrador.....	173
TABLA N° 3.16. Pruebas Funcionales. Administrador.....	174

RESUMEN

La presente tesis tiene como tema “ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”, la cual se desarrollo en el periodo Enero-Julio del 2012, y esta dirigido a estudiantes y docentes de la Carrera de Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.

Dicho trabajo tiene como objetivo el estudio y análisis de herramientas de distribución libre que sirvan para el desarrollo de aplicaciones en dispositivos móviles; tales como celulares u otros dispositivos que posean un Sistema Operativo Android.

Lo que conlleva a realizar el presente tema es la necesidad que existe en impulsar a docentes y estudiantes que deben aplicar nuevas tecnologías y específicamente en el desarrollo de aplicaciones móviles sobre un Sistema Operativo Android, y visto que también existe un desconocimiento total en el desarrollo de este tipo de aplicaciones.

Para incentivar en esta tecnología se desarrollo una aplicación de entretenimiento denominada “Mida su Inteligencia” en base a preguntas y respuestas de razonamiento, determinando al final del juego una puntuación que mide el grado de inteligencia que posee el jugador. Y dicha aplicación se encuentra disponible en la página web de la Universidad Tecnica de Cotopaxi Carrera de Ingenieria en Informatica y Sistemas Computacionales.

SUMMARY

The present thesis has a theme "ANALISIS AND STUDY OF FREE TOOLS FOR MOBILE APPLICATION DEVELOPMENT ON SMARTPHONES USING ADROID OPERATING SYSTEM AND THE CONSTRUCTION OF AN APPLICATION FOR ENTERTAINMENT OF THE USERS AT INFORMATIC AND COMPUTING SYSTEMS CAREER OF THE TECHNICAL UNIVERSITY OF COTOPAXI", which was developed in the period January-July 2012 and it is addressed to students and teachers of Informatics and Computing System Career of the University Technical of Cotopaxi.

This work has as objective the analysis and study of free distribution tools that works for the development of the application for mobile devices such as smartphones of any devices that have the operating system Android.

The reason to do this search is the existing need to impulse students and teachers who must to apply new technologies, specifically in the development of mobile applications on Android platform. It also important to mention that there is not a common knowledge about the development of this kind of applications.

To encourage this technology it developed an entertainment application called "Measure your Intelligence" it takes in base some questions and answers of logic, determining at the end of the game a score that measures the intelligence's grade that each player has on it. This application you can find available at Technical University of Cotopaxi's web page, Informatic and Computing Systems Career.

INTRODUCCIÓN

En estos últimos años gran parte del interés, tanto del sector académico como del comercial, se está moviendo hacia la distribución de contenidos sobre dispositivos móviles, motivado por las posibilidades que ofrecen estos terminales. Las nuevas características de los modernos terminales abren diferentes posibilidades para el aprendizaje a través de estos dispositivos, sin embargo el uso y desarrollo de aplicaciones de entretenimiento en los dispositivos móviles es poco habitual en nuestro medio.

Considerando los avances tecnológicos es el boom de los teléfonos inteligentes está llevando a fabricantes, desarrolladores y operadores a una auténtica carrera por lograr el desarrollo de las mejores aplicaciones móviles, y uno de los segmentos sobre los cuales se espera mayor impacto y crecimiento en el mercado de las aplicaciones móviles es el de las aplicaciones dirigidas al estudio y el entretenimiento.

La importancia de la tecnología móvil en nuestro medio y por ende las aplicaciones que se las instala como complemento de los medios tradicionales alcanza mayores niveles de eficiencia en el aprendizaje y entretenimiento estudiantil.

Actualmente la programación o desarrollo de aplicaciones sobre Android se ha convertido en un gran negocio y es una de las profesiones con más futuro y no sólo porque este sistema operativo este liderado por Google, sino porque es un beneficio para los desarrolladores realizar sus propias aplicaciones previo a esto haciendo un estudio de las herramientas.

Surge un gran problema en la Carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi se ha observado que en docentes y estudiantes existe un notable desconocimiento de cómo funciona el Hardware y Software en un dispositivo móvil y mucho menos utilizarlas correctamente o desarrollar aplicaciones para estos. La falta de investigación, desinterés en la información, hace que no aprovechemos la tecnología que tenemos a nuestro alcance y dentro de ello la tecnología móvil, esto ocasiona que los desarrolladores y/o usuarios queden al margen y sigan siendo solo consumidores de nuevas tecnologías acogiendo sin un análisis a estos productos pagando cantidades exorbitantes y desconociendo los costos reales de estos productos y servicios, es por esto que en la presente tesis se han trazado los siguientes objetivos, los cuales deben ser verificados para su cumplimiento:

Como objetivo general: “Realizar un análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles en teléfonos celulares con sistema operativo Android, y la construcción de una aplicación para entretenimiento de usuarios en la Carrera de Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi”.

El cual se desglosa en tres objetivos específicos que se llevaran a cabo para el desarrollo de cada capítulo:

- Analizar la información que permita tener bases teóricas para el desarrollo del tema de investigación propuesto.
- Recopilar la información de campo aplicando técnicas de investigación que permita conocer la situación actual de los usuarios que hacen uso de teléfonos celulares en la Universidad Técnica de Cotopaxi.
- Desarrollar una aplicación para entretenimiento de los usuarios de teléfonos celulares que se ejecute sobre una plataforma Android.

Cabe destacar que la hipótesis conlleva a una solución y es la siguiente: “El análisis y estudio de las herramientas libres permitirá el desarrollo y construcción de aplicaciones móviles en teléfonos celulares con sistema operativo Android en usuarios de celulares de la Carrera de Ingeniería en Informática y Sistemas Computacionales”.

Para llevar a cabo la presente tesis se tomó en cuenta la población de 309 personas entre docentes y estudiantes de la Carrera de Ingeniería en Informática y Sistemas Computacionales, dándonos como resultado de la muestra 73 nuestra población a investigar.

Los resultados obtenidos a través de la encuesta dirigida a docentes y estudiantes servirán como aporte investigativo para quienes deseen incursionar nuevas tecnologías este caso dispositivos móviles, de la misma manera desarrollar sus propias aplicaciones conociendo los pasos y que herramientas utilizar en el desarrollo de aplicaciones móviles bajo la plataforma Android realizando un previo análisis y estudio de herramientas libres para el desarrollo de las mismas.

Entonces la presente tesis se caracteriza por analizar y estudiar que herramientas son las adecuadas para el desarrollo de aplicaciones móviles ya que actualmente existen en el mercado un sin número de herramientas de diseño y programación, ya sean de distribución libre o propietarias.

La labor del análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles, es crear software para entretenimiento con su origen en la carrera de Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.

La presente tesis se encuentra distribuida en tres capítulos:

Capítulo I, Se analizó la información que permitió tener bases teóricas para el desarrollo del tema de investigación propuesto. Comprende los fundamentos teóricos que fortalecieron en el conocimiento del análisis y estudio de herramientas libres para el desarrollo de aplicaciones móviles, así como también un estudio bibliográfico actualizado sobre las fases que se debe cumplir para el proceso de desarrollo del software.

Capítulo II, Se recopiló información de campo aplicando técnicas de investigación que permitió conocer la situación actual de los usuarios que hacen uso de teléfonos celulares en la Universidad Técnica de Cotopaxi. Cuenta con información de la Institución y la Carrera donde se ejecuta toda la interpretación y análisis de resultados de esta investigación, además se considera gráficos estadísticos que darán un mejor entendimiento en la interpretación.

Capítulo III, Se desarrollo una aplicación para entretenimiento de los usuarios de teléfonos celulares que se ejecute sobre una plataforma Android. Se consideró la instalación y configuración de las herramientas a utilizarse en el desarrollo de la aplicación móvil, así como una metodología agil para el proceso de desarrollo de software, e incluye las conclusiones, recomendaciones y los aspectos de referencias bibliográficas.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

Según estudios, las personas accederán a aplicaciones de entretenimiento más a través de dispositivos móviles que a través de otros más convencionales como el PC. Con el desarrollo de aplicaciones para dispositivos móviles se ayudará a los usuarios a poseer esta tecnología en sus servicios principales o personales, dando esto la posibilidad y conveniencia de incrementar sus ingresos y acelerar sus procesos habituales.

La presente tesis explica el fundamento teórico básico y necesario para el análisis y estudio de las herramientas libres y la construcción de una aplicación para dispositivos móviles que se llevara a cabo en el presente trabajo de investigación propuesto. Para ello se utiliza las siguientes fuentes: libros, revistas, Internet y opiniones de expertos en el tema y experiencia en desarrollo de software para estos dispositivos.

1.2. Tecnologías de Comunicación Inalámbrica

Según MARTINEZ, Evelio en su obra publicada El ABC de las Telecomunicaciones manifiesta que “Las Tecnologías de Comunicación Inalámbricas o sin cables son aquellas en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. Los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.”2002 Pág. 58.

Puedo manifestar que en general las comunicaciones inalámbricas proveen beneficios adicionales que no proveen los sistemas cableados. El público o usuario final debe determinar cuales servicios son los más adecuados a sus necesidades, y desechar aquellos servicios que no le satisfagan.

Además es muy importante conocer sobre la tecnología que está detrás del servicio que el proveedor de servicios de telecomunicaciones nos brinda. Así se tendrá una visión más clara de las bondades o limitaciones, del servicio obtenido.

1.2.1. Historia

Según MARTINEZ, Evelio en su obra publicada El ABC de las Telecomunicaciones manifiesta que “En 1887 Heinrich Rudolph Hertz, demostró que existían las ondas electromagnéticas y que éstas podrían ser usadas para mover información a muy grandes distancias, además la unidad con la que son medidas las frecuencias del espectro electromagnético llevan su apellido (Hertz o Hz).

La base teórica de las ondas electromagnéticas fueron desarrolladas mucho antes por el físico escocés James Clerk Maxwell en 1864. El primer uso de las ondas electromagnéticas fue la telegrafía inalámbrica. Este relevante acontecimiento sería el predecesor de la propagación electromagnética o transmisión de radio.

Utilizando estos conceptos, el italiano Guglielmo Marconi inventa la radio en 1901. La radio fue el primer medio masivo de comunicación inalámbrica y a poco más de 100 años de su invención, las comunicaciones móviles han demostrado ser una alternativa a las redes cableadas para ofrecer nuevos servicios que requieren gran ancho de banda, pero con otros beneficios como la movilidad y la propagación de señales, estar comunicado en cualquier lugar, en cualquier momento.” 2002 Pág. 72.

Algunos de los beneficios que brindan las comunicaciones inalámbricas son las siguientes:

- Capacidad para un gran número de suscriptores.
- Compatibilidad a nivel nacional e internacional, para que los usuarios móviles puedan utilizar sus mismos equipos en otros países o áreas
- Prestación de servicios para aplicaciones de datos, voz y video.
- Adaptación a la densidad de tráfico; dado que la densidad de tráfico es diferente en cada punto de la zona de cobertura.
- Calidad del servicio en el caso de la voz comparable a servicio telefónico tradicional y accesible al público en general.

1.2.2. Clasificación de las Tecnologías de Comunicación Inalámbrica

Infrarrojo

- Estas redes son muy limitadas dado su corto alcance, necesidad de “Visión sin Obstáculos” entre los dispositivos que se comunican y su baja velocidad. Se encuentran principalmente en ordenadores portátiles, PDAs y algunas impresoras.

Bluetooth

- Es una tecnología orientada a la conectividad inalámbrica entre dispositivos estos pueden ser: computadoras de escritorio, PDAs, teléfonos móviles, auriculares, inclusive impresoras.

Wimax

- Wimax ofrece flexibilidad en la localización de las torres, gran alcance, un ancho de banda de hasta 50 veces el proporcionado por 3G. Wimax podría llevar internet a millones de hogares, empresas donde no llegan ni el cable ni las líneas de suscripción digital, especialmente en los países en desarrollo.

Wifi

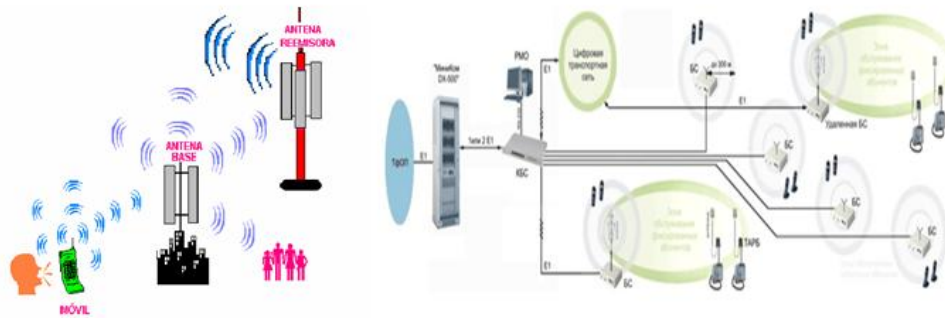
- La función de este tipo de redes es la de proporcionar conectividad y acceso a las tradicionales redes cableadas, pero con la flexibilidad y movilidad que ofrecen las comunicaciones inalámbricas.

Puedo manifestar que las tecnologías inalámbricas están teniendo mucho auge y desarrollo en estos últimos años, una de las que ha tenido un gran desarrollo ha sido la telefonía celular.

1.3. Telefonía celular

Según ASOCEL.ORG en el artículo telefonía celular publica que “La telefonía móvil celular es un servicio público de telecomunicaciones que le permite a todos los usuarios comunicarse desde el sitio en que se encuentren. Para poder lograr cobertura del servicio y comunicación desde cualquier sitio, se instalan estaciones distribuidas en toda la ciudad con antenas para atender zonas agrupadas en pequeñas celdas o células, y se deben instalar más estaciones a medida que aumentan los usuarios.”²² de Noviembre del 2011, disponible en la dirección web http://www.asocel.org/conferencias/conferencia_3.pdf

GRAFICO N° 1. 1. Telefonía celular



FUENTE. http://www.asocel.org.co/conferencias/conferencia_3.pdf

Pienso que los teléfonos celulares se han convertido en una herramienta primordial para la gente común y de negocios, las hace sentir más segura y las hace más productivas.

1.3.1. Breve historia

De acuerdo a EVELIUX.COM en su artículo la evolución de la telefonía móvil menciona que “Martin Cooper fue el pionero en esta tecnología, a él se le considera como “El Padre de la Telefonía Celular” al introducir el primer radio teléfono en 1973 en los Estados Unidos mientras trabajaba para Motorola; pero no fue hasta 1979 en que aparece el primer sistema comercial en Tokio Japón por la compañía NTT (Nippon Telegraph & Telephone Corp.)

En 1981 se introduce un sistema celular similar a AMPS (Advanced Mobile Phone System). Por otro lado, en los Estados Unidos gracias a que la entidad reguladora de ese país adopta reglas para la creación de un servicio comercial de telefonía celular, en octubre de 1983 se pone en operación el primer sistema comercial en la ciudad de Chicago.

La tecnología inalámbrica tuvo gran aceptación, por lo que a los pocos años de implantarse se empezó a saturar el servicio, por lo que hubo la necesidad de desarrollar e implementar otras formas de acceso múltiple al canal y transformar los sistemas analógicos a digitales para darle cabida a más usuarios.” 24 de Noviembre del 2011, disponible en la dirección virtual <http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-movil.php>

Puedo mencionar que la telefonía celular ha tenido un gran impacto, día a día en muchas personas nace el interés de tener en sus manos un celular, lo que implica que cada vez crecen los usuarios de estos teléfonos, exigiendo a desarrollar o implementar nuevas tecnologías para la comunicación móvil que satisfaga la necesidad de los usuarios.

En cuanto a las ventajas que ofrece la telefonía celular tenemos las siguientes:

1.3.2. Ventajas

Los usuarios se pueden comunicar en cualquier sitio que estén e igualmente pueden ser localizados por quien los necesite donde se encuentren, esto trae grandes beneficios para todos como por ejemplo:

- Para los padres de familia que necesitan localizar con urgencia al pediatra para que atienda a su hijo.
- Para los periodistas que necesitan de inmediato dar o recibir una noticia donde se encuentren.
- Para el que necesita ser auxiliado en una emergencia o requiere dar o recibir información urgente.

La telefonía celular se ha categorizado por generaciones. A continuación la dirección antes mencionada describe cada una de ellas:

1.3.3. Generaciones de la Telefonía Móvil

1.3.3.1. La primera generación 1G

La 1G de la telefonía móvil hizo su aparición en 1979, se caracterizó por ser analógica y estrictamente para voz. La calidad de los enlaces de voz era muy baja, baja velocidad, la transferencia entre celdas era muy imprecisa, tenían baja capacidad (basadas en FDMA, Frequency Division Multiple Access) y la seguridad no existía. La tecnología predominante de esta generación es AMPS (Advanced Mobile Phone System).

1.3.3.2. La segunda generación 2G

La 2G arribó hasta 1990 y a diferencia de la primera se caracterizó por ser digital. El sistema 2G utiliza protocolos de codificación más sofisticados y son los sistemas de telefonía celular usados en la actualidad. La 2G ó (Personal Communications Services).

Las tecnologías predominantes son: GSM (Global System for Mobile Communications), CDMA (Code Division Multiple Access) y PDC (Personal Digital Communications), éste último utilizado en Japón.

Los protocolos empleados en los sistemas 2G soportan velocidades de información más altas para voz pero limitados en comunicaciones de datos. La mayoría de los protocolos de 2G ofrecen diferentes niveles de encriptación.

1.3.3.3. La tercera generación 3G

La 3G es tipificada por la coincidencia de la voz y datos con acceso inalámbrico a Internet, aplicaciones multimedia y altas transmisiones de datos. Los protocolos empleados en los sistemas 3G soportan altas velocidades de información enfocados para aplicaciones más allá de la voz tales como audio (MP3), video en movimiento, video conferencia y acceso rápido a Internet, etc.

Los sistemas 3G alcanzan velocidades de hasta 384 Kbps permitiendo una movilidad total a usuarios viajando a 120 kilómetros por hora en ambientes exteriores y alcanzará una velocidad máxima de 2 Mbps permitiendo una movilidad limitada a usuarios caminando a menos de 10 kilómetros por hora en ambientes estacionarios de corto alcance.

Entre las tecnologías de la tercera generación se encuentran UMTS (Universal Mobile Telephone Service), entre otras.

1.3.3.4. La cuarta generación 4G

La cuarta generación es un proyecto a largo plazo que será 50 veces más rápida en velocidad que la tercera generación. Desde el año 2005 se empezaron hacer pruebas con esta tecnología y a mediados del año 2010 se comercializaron estos servicios con tecnología 4G.

Según las generaciones de la telefonía móvil se puede decir que no es más que el avance o nuevas técnicas que se implementan en un determinado periodo de tiempo para satisfacer las necesidades de usuarios de celulares, es así que hasta el día de hoy existe hasta la cuarta generación, lo que se puede evidenciar que este tipo de tecnologías crecen a un ritmo muy acelerado y que brindan mucho mas servicios a usuarios sin importar edad o género.

1.4. *Dispositivos Móviles*

De acuerdo a GARCÍA, Rosana, en su obra Dispositivos Móviles manifiesta que “Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente a una red, con memoria limitada, ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales como transmitir señales acústicas por medio de señales eléctricas a distancia.”2008, Pag.01.

Puedo manifestar que los dispositivos móviles o electrónicos de comunicación poseen las mismas capacidades básicas que un teléfono de línea convencional, además de su propiedad de movilidad es de tipo inalámbrico ya que no requiere cables conductores para lograr su conexión a la red telefónica.

1.5. Software Libre

Conforme a HISPALINUX.ES en su artículo que es el software libre publica que “Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. En el caso de la distribución, puede ser licencia tipo BSD (libertad de distribución a código cerrado) o GPL (distribución total, pero bajo las condiciones de tener el código abierto).”²⁶ de Noviembre del 2011, disponible en la dirección web <http://www.hispalinux.es/SoftwareLibre>

El Presidente de la República, Rafael Correa, mediante decreto No. 1014 de 10 de abril del 2008, establece como política pública para las entidades de la administración pública central la utilización de Software Libre en sus sistemas y equipamientos informáticos.

Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan su acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas. Al utilizar software libre tendríamos acceso a tecnología de primer nivel, herramientas antes inaccesibles para el trabajo, para los negocios y para los estudios y se enriquecería el conocimiento de la población.

Para que un programa sea considerado bajo software libre debe cumplir cuatro libertades específicas estos se detallan a continuación:

- La libertad de usar el programa, con cualquier propósito (**libertad 0**).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (**libertad 1**).
- La libertad de redistribuir copias, para que pueda ayudar a sus amigos (**libertad 2**).

- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (**libertad 3**).

Respecto a software libre considero que un programa es software libre si los usuarios tienen todas las libertades. Se debería tener la libertad de distribuir copias, sea con o sin modificaciones, gratis o cobrando una cantidad por la distribución. De la misma manera que el software propietario el software libre también tiene sus licencias de uso.

1.5.1. Licencias GPL

De acuerdo a WIKIPEDIA.ORG en su artículo software libre manifiesta que “Una de las más utilizadas es la Licencia Pública General de GNU (GNUGPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNUGPL.

1.5.2. Bondades de software libre

Bajo costo de adquisición: Se trata de un software económico que permita un ahorro de grandes cantidades en la adquisición de las licencias.

Innovación tecnológica: Cada usuario puede aportar sus conocimientos y su experiencia de manera conjunta hacia donde se debe dirigir la evolución y el desarrollo del software.

Independencia del proveedor: Al disponer del código fuente, se garantiza una independencia del proveedor que hace que cada empresa o particular pueda seguir contribuyendo al desarrollo y los servicios del software.

Escrutinio público: La corrección de errores y la mejora del producto debe llevarse a cabo de manera rápida y eficaz por cada usuario que llegue a utilizar el producto.

Adaptación del software: Esta cualidad resulta de gran utilidad para empresas e industrias específicas que necesitan un software personalizado para realizar un trabajo específico y con software libre.

Lenguas: Aunque el software se cree y salga al mercado en un solo idioma, el hecho de ser software libre facilita en gran medida su traducción y localización para que usuarios de diferentes partes del mundo puedan aprovechar estos beneficios.” 24 de Noviembre del 2011, disponible en la web http://es.Wikipedia.org/wiki/Software_libre

Una vez entendido que es software libre y que se debe tener en cuenta para desarrollar software ya sea web de escritorio y para dispositivos móviles se procederá a estudiar que es un sistema operativo para que sirve y cuales se acoplan en los dispositivos móviles.

1.6. Sistema Operativo

Según DEFINICION.DE en su artículo definición de sistema operativo menciona que “El conjunto de programas informáticos que permite la administración eficaz de los recursos de una computadora es conocido como sistema operativo o software de sistema. Estos programas comienzan a trabajar apenas se enciende el equipo, ya que gestionan el hardware desde los niveles más básicos y permiten además la interacción con el usuario.” 26 de Noviembre del 2011, disponible en la dirección virtual <http://definicion.de/sistema-operativo/>

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

A continuación se menciona algunos de los sistemas operativos que actualmente vienen acoplados en los dispositivos electrónicos (teléfonos celulares).

1.6.1. Tipos de Sistemas Operativos para dispositivos móviles

1.6.1.1. iOS

Al consultar BLOGS.RPP.COM.PE en su artículo Sistemas Operativos para Celulares menciona que “iOS es el sistema operativo desarrollado por Apple para sus dispositivos móviles iPhone y iPad. Originalmente fue diseñado para iPhone, pero después se utilizó para otros dispositivos de Apple como (iPad o el iPod Touch). El sistema de distribución de aplicaciones es el App Store, el cual contiene actualmente más de 250.000 aplicaciones. Actualmente está considerada como la plataforma que más aplicaciones posee.

Al igual que Google, Apple saca actualizaciones de su sistema operativo para dispositivos móviles, aunque no con tanta frecuencia como Android. Su sistema operativo soporta la tecnología multitáctil y la multitarea. Su principal punto débil es que no soporta Flash ni Java y es muy dependiente de la aplicación iTunes, para gestionar el contenido del dispositivo.

1.6.1.2. Android

Android es un sistema operativo diseñado para dispositivos móviles, como teléfonos celulares (HTC, Samsung, Motorola, Tablet, PC's, netbooks). Es un sistema operativo abierto diseñado por la Open Handset Alliance (OHA), basado en el kernel de Linux. Es multitarea, soporta la tecnología multitáctil y permite el uso de módulos de accesibilidad. El sistema de distribución de aplicaciones es el Android market.

Las aplicaciones para Android pueden escribirse tanto en Java como en C/C++. Continuamente salen al mercado actualizaciones del sistema operativo, con las que van aumentando su funcionalidad y corrigiendo fallos del sistema operativo.

1.6.1.3. RIM OS

Rim es el sistema operativo desarrollado por la compañía (RIM) Research In Motion para las BlackBerry. Soporta la multitarea y el lenguaje de programación que utiliza es Java. El sistema de distribución de aplicaciones para este sistema operativo es BlackBerry App World. La aplicación estrella de este sistema operativo, es BBM que habilita la posibilidad de mandar mensajes a otra BlackBerry, mediante el PIN único que posee cada BlackBerry por la seguridad de estos dispositivos.

Gracias a que llevan cinco años más en el mercado que las dos plataformas anteriores, son uno de los dispositivos más interesantes en ambientes laborales, donde se precisa escribir mucha cantidad de mensajes y una alta movilidad.

1.6.1.4. Symbian OS

Symbian es el sistema operativo diseñado para dispositivos móviles Nokia, creado por diferentes compañías de empresas de telefonías móviles. Su sistema operativo soporta la multitarea y posee como lenguaje de programación para desarrollar aplicaciones para el sistema operativo, C++. Aún así es posible crear aplicaciones en Java pues tiene integrada una máquina virtual en su sistema.

El sistema de distribución de aplicaciones es el Nokia OviStore, para descargar aplicaciones. Al igual que otros sistemas operativos para móviles, posee diferentes versiones para su sistema operativo. Ha perdido cuota de mercado con iOS y Android, pero con la llegada de su nuevo sistema operativo MeeGo espera recuperarse.

1.6.1.5. Windows Phone

Windows Phone, anteriormente llamado Windows Mobile, es el sistema operativo desarrollado por Microsoft para los dispositivos móviles (HTC 7 pro, LG Quantum, etc). Soporta la multitarea y su plataforma de desarrollo es silverlight.

Para distribuir sus aplicaciones posee el Windows Market place for Mobile. Al igual que otros sistemas operativos, posee diferentes versiones, siendo la última versión anunciada Windows Phone 7.” 28 de Noviembre del 2011, disponible en la web <http://blogs.rpp.com.pe/technovida/2011/03/15/%C2%BFde-que-se-trata-todo-esto-de-android-en-mi-telefono/>

Considero que un sistema operativo es el que da vida a un dispositivo móvil (hardware), con el cual logramos administrar, controlar y ejecutar a todos los periféricos de entrada, salida y almacenamiento del equipo electrónico.

Cabe destacar que los sistemas operativos no funcionan sólo en las PC's por el contrario, los sistemas operativos se encuentran en la mayoría de dispositivos electrónicos que utilizan microprocesadores: el software de sistema posibilita que el aparato cumpla con sus funciones.

Después de conocer los conceptos de sistema operativo y cuales vienen acoplados para cada tipo de celular se realizara un análisis de herramientas para desarrollar aplicaciones móviles sobre un celular con sistema operativo android.

1.7. Análisis de Herramientas

El desarrollo de aplicaciones móviles en la actualidad es conocido por la mayor parte de usuarios y profesionales que crean sistemas de información, especialmente por quienes se dedican al estudio de nuevas tendencias tecnológicas, mas aun siendo estudiantes de Informatica se debería adentrar no solo en el desarrollo de software web o escritorio sino buscar la manera de aprender a programar para dispositivos móviles.

A la hora de escoger herramientas sofisticadas de programacion ya sea para software web, escritorio o para dispositivos móviles se debe realizar un análisis previo de dichas herramientas considerando ventajas y limitaciones que disponen las mismas para luego ser usadas en el proceso de desarrollo.

A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos actualmente es la siguiente:

1. **ANDROID** 68,1%
2. **iOS** 16,9%
3. **BlackBerry OS** 4,8%
4. **Symbian OS** 4,4%
5. **Windows Phone y Windows Mobile** 3,5 %
6. **Linux u otros** 2,3%

Android tiene la mayor cuota, desde enero del 2011, con más de la mitad del mercado, experimentó un creciente aumento a inicios del 2009 hasta hoy actualmente es el sistema operativo móvil más utilizado.

Para escoger las mejores herramientas y el software apropiado de desarrollo de aplicaciones se recurrió a la información en internet, en libros, revistas, opiniones de expertos en el tema de desarrollo de aplicaciones móviles bajo un sistema operativo apropiado, se puede manifestar que existen pocas en el mercado pero muy sofisticadas a la hora de programar y desarrollar aplicaciones.

Entre las herramientas se hizo un análisis previo de cuales son las apropiadas para el desarrollo con el sistema operativo Android. Se analizó entre ellos lenguajes de programación, ides de programación, bases de datos, maquinas virtuales, etc. (**VER ANEXOS N° 3.2 A 3.6**)

Después de todo lo mencionado anteriormente amerita que a continuación se haga un estudio de las herramientas libres para el desarrollo de aplicaciones móviles, teniendo ya definidas cuales se utilizaran y que sirvan de apoyo en el desarrollo de la presente tesis.

1.8. Estudio de Herramientas libres

Para el desarrollo de aplicaciones en teléfonos celulares tenemos diferentes herramientas que cumplen funciones específicas ya sea como máquina virtual, base de datos, emuladores, y/o lenguajes de programación etc.

Para iniciar a programar bajo cualquier sistema operativo en este caso Android se debe empezar por el estudio de un buen lenguaje de programación. Y se escogió el lenguaje de programación Java.

1.8.1. Lenguajes de Programación

Según DEFINICIÓN.ORG en su artículo Definición de Lenguaje de Programación manifiesta que “Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.” 28 de Noviembre del 2011, disponible en el internet <http://www.definicion.org/lenguaje-de-programacion>

Pienso que un lenguaje de programación está diseñado para expresar instrucciones que pueden ser llevadas a cabo por máquinas como las computadoras. Se puede usar para crear programas que controlen el comportamiento físico y lógico de un equipo electrónico, para expresar algoritmos con precisión, o como modo de comunicación humana.

1.8.1.1. Lenguaje Java

De acuerdo a JAVA.COM en su artículo Lenguaje Java se encuentra que: “Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en

más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión

Lenguaje Java entendemos por la capacidad de crecimiento y difusión que tiene el código para que funcione sobre cualquier plataforma de software y hardware. Esto significa que el mismo programa escrito para Linux pueda ser ejecutado en Windows sin ningún problema.

Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras.”05 de Diciembre del 2011, disponible en la dirección web http://java.ciberaula.com/articulo/que_es_java/.

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general y más aun tratándose de desarrollar aplicaciones para celulares, java es muy versátil y eficiente en su uso. Por todo esto se eligió utilizar el lenguaje java en desarrollo de la presente tesis.

1.8.1.1.1. Características

De acuerdo a las características del lenguaje java el sitio web citado anteriormente menciona las siguientes:

Desarrollar software en una plataforma y ejecutarlo en otra.

- Crear programas para que funcionen en un navegador Web y en servicios Web
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios basados en la tecnología Java para crear servicios o aplicaciones totalmente personalizados

- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y cualquier dispositivo digital.
- "Write once, run everywhere" (Escribalo una vez, ejecutelo en todos lados).

1.8.1.2. Lenguaje XML

De acuerdo a DEFINICION.DE en su artículo publicado definición de XML menciona que “XML proviene de Extensible Markup Language (Lenguaje de Marcas Extensible). Se trata de un metalenguaje (un lenguaje que se utiliza para decir algo sobre otro lenguaje) extensible de etiquetas que fue desarrollado por el World Wide Web Consortium (W3C), un consorcio internacional que elabora recomendaciones para la World Wide Web.

El XML es una adaptación del SGML (Standard Generalized Markup Language), un lenguaje que permite la organización y el etiquetado de documentos. Esto quiere decir que el XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades.” 05 de Diciembre del 2011, disponible en la página web <http://definicion.de/xml/>

GRAFICO N° 1. 2. Ejemplificación del lenguaje de marcas extendido XML

```

Feeds.xml  RSSReader_CSharp.aspx.cs  RSSReader_CSharp.aspx
1  <?xml version="1.0" encoding="utf-8" ?
2
3  <Feeds>
4
5  <Feed>
6      <Name>ScottGu Blog</Name>
7      <Url>http://weblogs.asp.net/sc
8  </Feed>
9
10 <Feed>
11     <Name>weblogs.ASP.NET Main Fee
12     <Url>http://weblogs.asp.net/Ma
13 </Feed>
14
  
```

FUENTE. <http://definicion.de/xml/>

1.8.1.2.1. Ventajas

De acuerdo a WIKIPEDIA.COM en su artículo XML menciona las siguientes ventajas“

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear uno para cada versión de XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. Sirven para evitar *bugs* (errores de compilación) y se acelera el desarrollo de aplicaciones.
- Si alguien utiliza un documento creado en XML, es sencillo entender su estructura y procesarla. Se puede comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos.” 08 de Diciembre del 2011, disponible en la web [http://es.wikipedia.org/wiki /Extensible_Markup_Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language).

XML tiene 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. La facilidad de trabajar con XML es que intercambia información estructurada en una base de datos, editores de texto y hojas de cálculo.

1.8.2. Programacion Orientada a Objetos

De acuerdo a HERNANDEZ Carlos, en su obra publicada en Slideshared titulada Programación Orientada a Objetos menciona que “La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Esta basada en el modo de pensar del hombre y no al modo de pensar de la maquina.

La orientación a objetos es un paradigma de programación que facilita la creación de software de calidad por sus factores que potencian el mantenimiento, la extensión y la reutilización del software generado bajo este paradigma.

El elemento básico de este paradigma no es la función (elemento básico de la programación estructurada), sino un ente denominado objeto. Un objeto es la representación de un concepto para un programa, y contiene toda la información necesaria para abstraer dicho concepto: los datos que describen su estado y las operaciones que pueden modificar dicho estado, y determinan las capacidades del objeto.”2009, Pág.02

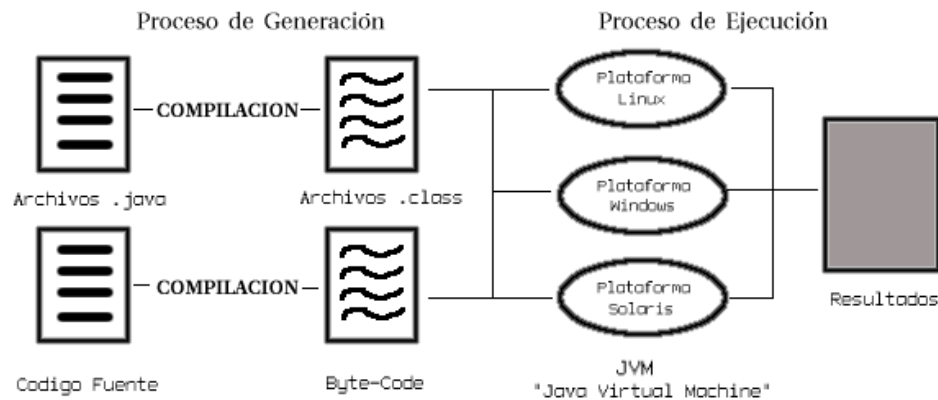
Una vez conocido que lenguaje de programación se usara y su manera de trabajar se procederá a estudiar todas las herramientas necesarias para iniciar a programar o desarrollar aplicaciones sobre la misma. Se iniciara con una maquina virtual indispensable de java.

1.8.3. JVM

Según GLOBALMENTORING en su artículo JVM, JDK, JRE Conceptos Fundamentales de la P.O.O. menciona que “JVM o Maquina Virtual de Java o sus siglas en ingles (Java Virtual Machine) es la aplicación donde corren los programas hechos en Java, es nativa del sistema operativo y usualmente viene cargada por default en el pc. Con esta herramienta no es posible desarrollar, solo pueden desplegarse aplicaciones. De ahí el axioma de escríbelo una vez y ejecútalo en varias partes o “Write once, run anywhere”.

JVM es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.” 08 de Diciembre del 2011, disponible en la web <http://globalmentoring.com.mx/cursos-java/java-fundamentos/diferencias-entre-jdk-jre-y-jvm>

GRAFICO N° 1. 3. Proceso de generación y ejecución de obtención de resultados



FUENTE. <http://javabasico.osmosislatina.com/curso/objetos.htm>

1.8.4. JRE

Según GLOBALMENTORING en su artículo JVM, JDK, JRE Conceptos Fundamentales de la P.O.O. menciona que “El JRE es un Ambiente de Recorrido de Java o sus siglas en inglés (Java Runtime Environment) es un conjunto de utilidades de Java contiene la JVM. Con esta herramienta tampoco es posible el desarrollo o la programación únicamente es el proceso del sistema operativo que permite correr la aplicación.” 08 de Diciembre del 2011, disponible en la web <http://globalmentoring.com.mx/cursos-java/java-fundamentos/diferencias-entre-jdk-jre-y-jvm>

1.8.5. JDK

Según GLOBALMENTORING en su artículo JVM, JDK, JRE Conceptos Fundamentales de la P.O.O. manifiesta que “JDK es un Kit de Desarrollo de Java o sus siglas en inglés (Java development kit), es un software que provee herramientas de desarrollo para la creación de programas en java.

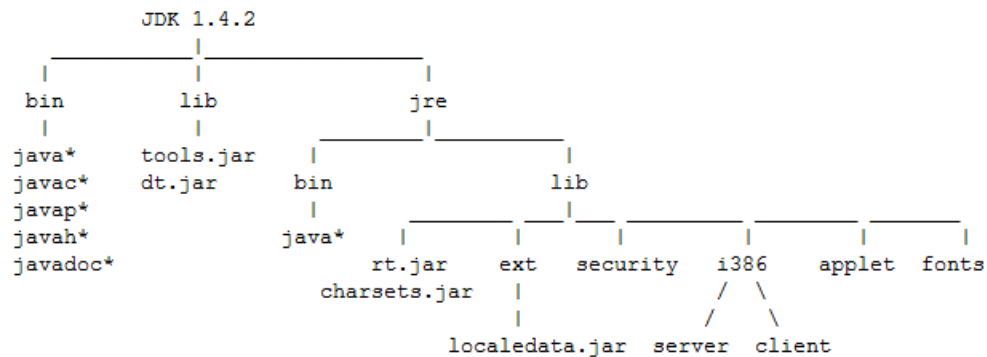
El JDK es el kit para desarrolladores, contiene entre otras cosas el JRE y la JVM. JRE actúa como un “intermediario” entre el sistema operativo y el lenguaje Java.

Si se quiere desarrollar una aplicación para Android se tendría que bajar el SDK y configurarlo en el IDE eclipse (en el que empieza la programación). El SDK es para android lo que es el JDK para Java.” 08 de Diciembre del 2011, disponible en la web <http://globalmentoring.com.mx/cursos-java/java-fundamentos/diferencias-entre-jdk-jre-y-jvm>

1.8.5.1. Estructura JDK

La estructura de directorios del JDK es la siguiente:

GRAFICO N° 1. 4. Estructura de directorios Jdk



FUENTE. <http://javabasico.osmosislatina.com/curso/objetos.htm>

Como se puede observar, la estructura se encuentra dividida en tres grandes partes:

- **bin:** Incluye los ejecutables para la generación de programas Java.
- **lib:** Contiene las librerías (Clases) base empleadas en la generación de la gran mayoría de programas Java.
- **jre:** Incluye el ambiente necesario para ejecutar programas Java, el cual a su vez se encuentra sub-dividido en distintos directorios.

1.8.6. Android SDK

De acuerdo a RAMIRES Enrique, en su obra Desarrollo de Aplicaciones para Dispositivos con Sistema Operativo Android manifiesta que “Android SDK es un kit de desarrollo de software de las siglas en inglés (Software Development kit) para android consta de un conjunto de herramientas para el desarrollo que permite a los desarrolladores y programadores confeccionar aplicaciones y juegos para dispositivos móviles.”2009, Pág. 10

Android SDK se trata de una interfaz de programación de aplicaciones (API application programming interface) que hace uso del lenguaje de programación Java permite su ejecución tanto en sistemas Linux, Windows o Mac

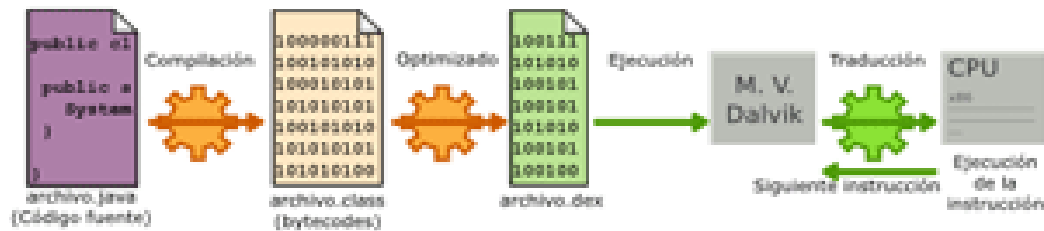
1.8.6.1. Máquina Virtual Dalvik

Según ANDROIDEITY.COM en su artículo Máquina Virtual Dalvik menciona que “Dalvik es la máquina virtual que utiliza la plataforma para dispositivos móviles Android. Dalvik ha sido diseñada por Dan Bornstein con contribuciones de otros ingenieros de Google.

Todas las aplicaciones se programan en el lenguaje Java y se ejecutan mediante una máquina virtual Dalvik, específicamente diseñada para Android. Esta máquina virtual ha sido optimizada y adaptada a las peculiaridades propias de los dispositivos móviles (menor capacidad de proceso, baja memoria, alimentación por batería, etc.) y trabaja con ficheros de extensión .dex (Dalvik Executables). ” 08 de Diciembre del 2011, disponible en la web <http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>

Dalvik no trabaja directamente con el bytecode de Java, sino que lo transforma en un código más eficiente que el original, pensado para procesadores pequeños.

GRAFICO N° 1.5. Codificación eficiente basada en procesadores pequeños para dispositivos



FUENTE. <http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>

Puedo manifestar que una máquina virtual aunque no se la pueda ver físicamente sabemos que esta instalada porque sirve para poder ejecutar los programas o aplicaciones que realicen actúa como un software que emula a una computadora y puede ejecutar programas como si fuese una computadora real.

1.8.6.2. Emulador Android AVD/ Dalvik

Según RAMIREZ, Enrique, en su obra Desarrollo de Aplicaciones para Dispositivos con Sistema Operativo Android menciona que "El emulador de Android está contenido dentro del paquete del SDK, se trata de una herramienta creada por Google para poder probar las aplicaciones móviles sin necesidad de instalarlas en un dispositivo, mediante el proceso de creación de un proyecto en Eclipse se hará uso de uno de los terminales emulados creados para probar en cada momento las nuevas funciones o modificaciones de las aplicaciones.

Este emulador incluye diferentes skins (Elemento que permite cambiar la apariencia externa de la interfaz de una aplicación sin afectar su funcionalidad) para representar gráficamente un dispositivo móvil real, con pantalla, teclado y botones de función, así como aplicaciones de usuario preinstaladas que incluyen navegador web o visor de mapa, y una consola que permite interactuar con él a través de Telnet." 2009, Pág. 10

Puedo manifestar que mediante la utilización de un emulador con esta función se puede crear un dispositivo móvil casi completamente funcional dentro del sistema operativo, PC o Mac, para el desarrollo de las aplicaciones. Este emulador es más conocido dentro el paquete del sdk como AVD (Dispositivo Virtual de Android).

1.8.6.3. ADT

De acuerdo a ANDROID DEVELOPERS en su artículo publicado Android Developer Tools manifiesta que “ADT (Android Developer Tools) es un plugin para Eclipse que proporciona un conjunto de herramientas que se integran con el IDE de Eclipse. Ofrece acceso a muchas características que ayudan a desarrollar aplicaciones Android rápidamente. ADT proporciona acceso GUI a muchas de las herramientas de línea de comandos del SDK, así como una herramienta de diseño de interfaz de usuario para creación rápida de prototipos, el diseño y la construcción de la interfaz de usuario de la aplicación, agrega paquetes basados en la API de Android Framework, incluso exporta archivos **.apk** con el fin de distribuir la aplicación.”¹² de Enero del 2012, disponible en la web <http://developer.android.com/tools/help/adt.html>

1.8.6.4. Base de Datos

Una base de datos abarca una sección o conjunto de datos que pertenecen a un mismo contexto estos datos están ordenados y almacenados sistemáticamente para usarlos posteriormente.

1.8.6.4.1. SQLite

Según APLICACIONESEMPRESARIALES.COM en el artículo Base de Datos Sqlite señala que “SQLite es un motor de base de datos relacional gratis y de código abierto. A diferencia del sistema de gestión de base de datos cliente-servidor, esta no necesita de un servidor para comunicarse con los datos sino que es directamente entre la aplicación y los datos, esta es portable y puede ser embebida en cualquier tipo de software, ya sea web, de escritorio o para tecnologías móviles. ” 12 de Diciembre del 2011, disponible en la web <http://www.Aplicaciones empresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>

Puedo manifestar que es una base de datos totalmente confiable, accesible, que puede ser acoplada en múltiples tipos de proyectos y lenguajes, permitiendo trabajar con múltiples tablas, índices, triggers y vistas, y todo contenido en un solo archivo, además es totalmente independiente del sistema operativo en que se utilice.

1.8.7. Ide's de Programación

Un IDE o sus siglas en inglés (Integrated Development Environment) es una aplicación compuesta por un conjunto de herramientas útiles para un programador, puede ser exclusivo para un lenguaje de programación consiste de un editor de código, un compilador, un debugger y un constructor de interfaz gráfica GUI.

1.8.7.1. Eclipse

De acuerdo a ECURED en su artículo Eclipse manifiesta que “Eclipse es un (IDE) Entorno de Desarrollo Integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse.

Eclipse es un IDE para el desarrollo de software de código abierto está escrito totalmente en Java permite crear aplicaciones empresariales, móviles, etc. Una de sus principales ventajas es que además de ser multiplataforma es también multilenguaje. Permite desarrollar aplicaciones escritas en Java, C++, PHP, Ruby, Python, Javasript, etc.” 12 de Diciembre del 2011 disponible en el internet <http://www.cursosporinternet.info/index.php/the-news/43-programacion/167-los-ides-de-programacion.html>

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL).

Pienso que actualmente eclipse es el ide más reconocido y utilizado por la mayoría de programadores en java por su facilidad de uso sus entornos gráficos, además forma parte de software libre y su costo de obtención de este software no tiene precio pero se debe saber cómo utilizar al momento de su programación.

Características de Eclipse

- Eclipse dispone de un Editor de texto con resaltado de sintaxis.
- La compilación es en tiempo real.
- Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.

1.8.8. Sistema operativo Android

1.8.8.1. Historia

Según RAMIREZ Enrique, en su obra desarrollo de aplicaciones para dispositivos menciona que “En Julio de 2005, Google adquirió Android, una pequeña empresa de California. En esos momentos, la compañía se dedicaba a la creación de software para teléfonos móviles. Una vez en Google, el equipo desarrolló un sistema operativo basado en Linux para dispositivos móviles. Más adelante, Google adaptó su buscador y sus aplicaciones para su uso en móviles.

En septiembre del 2007, Google tenía varias patentes de aplicaciones sobre el área de la telefonía móvil. El 5 de noviembre del mismo año, se anunció la fundación de la Open Handset Alliance al mismo tiempo que la creación de la plataforma Android.

La Open Handset Alliance está formada por un consorcio de 34 compañías de hardware, software y telecomunicaciones, entre las cuales se incluyen Google, HTC, Intel y Motorola entre otras, dedicadas a investigar estándares abiertos para dispositivos móviles.

El primer teléfono en el mercado que posee Android es el T-Mobile G1(también conocido como Dream), que viene con la versión Android 1.0 preinstalada. Este móvil es el resultado conjunto de T-Mobile, HTC y Google. Desde el 21 de octubre de 2008, android está disponible como código abierto.”2009, pág. 15

Pienso que es muy importante conocer la historia de un sistema operativo quien lo creo y como se puede utilizar para futuras aplicaciones, ya que asi se conocera bondades y limitaciones de un sistema operativo, de esta manera nos familiarizaremos mas con la plataforma. A continuación breves definiciones del sistema operativo Android

1.8.8.2. Definiciones

Conforme a RAMIRES Enrique, en su obra Desarrollo de Aplicaciones para Dispositivos con Sistema Operativo Android manifiesta “Algunas de las definiciones de android:

- Android es un conjunto de herramientas y aplicaciones vinculadas a una distribución Linux para dispositivos móviles.
- Android es de código abierto, gratuito y no requiere pago de licencias.
- Android es una plataforma de código abierto para dispositivos móviles que está basada en Linux y desarrollada por Open Handset Alliance.
- Es un paquete de software para dispositivos móviles que incluye un sistema operativo, actúa como intermediario para comunicarse con otras aplicaciones. Los desarrolladores pueden crear aplicaciones para la plataforma usando el SDK de android.

Las solicitudes se han escrito utilizando el lenguaje de programación Java y se ejecutan en Dalvik, una máquina virtual personalizada que se ejecuta en la parte superior de un núcleo de Linux la cual es diseñada para uso en dispositivos móviles.” 2009, pag.16

1.8.8.3. Características de Android

Así mismo el autor antes mencionado hace referencia algunas de las características de android que son las siguientes:

- Amplia variedad de diseños (librerías de gráficos 2D y 3D)
- Almacenamiento de datos en BBDD SQLite
- Conectividad (CDMA, Bluetooth y Wi-Fi)
- Mensajería (SMS y MMS)
- Navegador Web
- Máquina virtual de Java
- Soporte de formatos (MPEG-4, MP3, JPG, PNG, GIF)
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse).

1.8.8.4. Arquitectura de Android

Los componentes del sistema operativo de Android, cada sección se describe en detalle a continuación:

Aplicaciones: Las aplicaciones base incluyen un cliente de email, programa de SMS, calendario, aplicaciones, navegador, contactos, y otros. Todas las aplicaciones son escritas en el lenguaje de programación Java.

Framework de aplicaciones: Los desarrolladores tienen acceso completo a los APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes.

Una capa de servicios disponibles para las aplicaciones incluye:

- Un completo conjunto de vistas, pueden ser utilizadas para desarrollar una aplicación: listas, cajas de texto, botones, etc.
- Proveedores de contenidos, permiten el acceso a datos provenientes de otras aplicaciones (cómo Contactos), comparten sus propios datos.
- Administrador de recursos, provee acceso a recursos como cadenas, gráficos y archivos.
- Administrador de notificaciones, permite a todas las aplicaciones mostrar alertas personalizables en la barra de estatus.
- Administrador de actividades, maneja el ciclo de vida de las aplicaciones y provee un comportamiento común en la navegación.

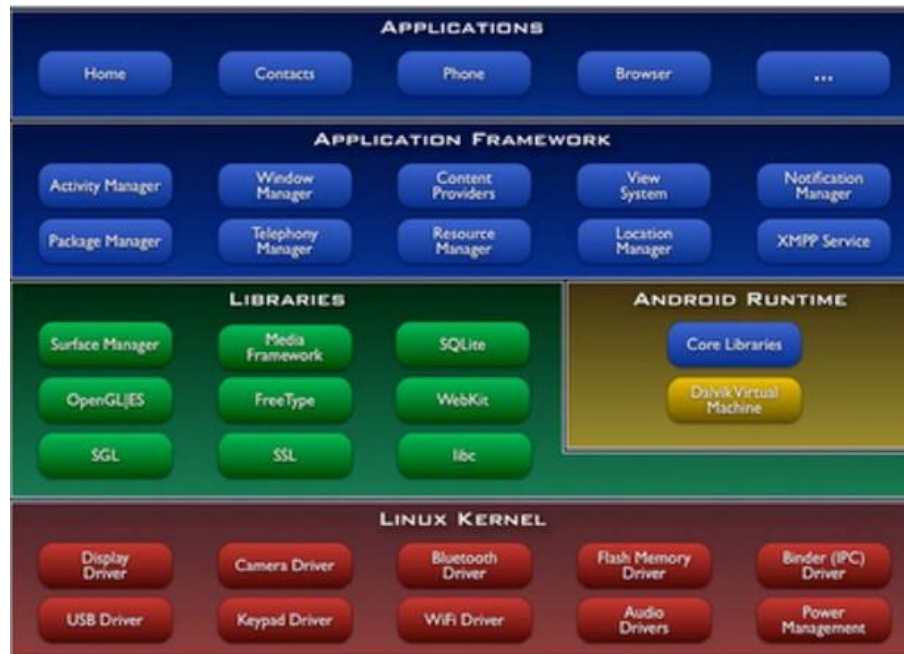
Librerías: Android incluye un conjunto de librerías C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library, librerías de medios, librerías de gráficos3d, SQLite, entre otras.

Runtime de Android: Android incluye un conjunto de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima.

Núcleo - Linux: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria y procesos, stack de red, y modelo de drivers. El núcleo actúa como una capa de abstracción entre el hardware y el software.

El siguiente diagrama muestra los principales componentes del sistema operativo Android:

GRAFICO N° 1.6. Arquitectura del Sistema Operativo Android



FUENTE. <http://riunet.upv.es/bitstream/handle/10251/10299/Memoria.pdf>

1.9. Metodología de Desarrollo

Para que un proyecto de investigación sea satisfactorio sin importar el tipo de software ya sea para dispositivos móviles, escritorio o web es necesario que el resultado cuente con la calidad esperada por el cliente, usuarios y el desarrollador. La calidad del proyecto se puede definir de acuerdo al desempeño del desarrollador, de las herramientas utilizadas y de la metodología aplicada.

Considero que la metodología es una guía para el desarrollador que siga paso a paso todo el proceso de desarrollo de software. Sin una metodología ésta presente tesis no tendría inicio ni fin, por tal razón se cree apropiado aplicar las fases de una metodología ágil como es la XP.

Las Metodologías de Desarrollo de Software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelamiento de sistemas que permitan desarrollar software de calidad.

1.9.1. Metodologías ágiles

Según AGILEDOCTEMASANV en su artículo Metodologías Agiles menciona que “Las metodologías ágiles permiten a los equipos desarrollar software rápidamente y responder a los cambios que pueden surgir a lo largo del proyecto o durante cada una de las fases. Esto pretende ser una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas.

El desarrollo ágil de software promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Desarrollando software en cortos lapsos de tiempo se minimizan los riesgos, cada una de esas unidades de tiempo se llama iteración.

Cada iteración no debe añadir demasiada funcionalidad, para justificar el lanzamiento del producto al mercado, sino que la meta debe ser conseguir una versión funcional sin errores.” 12 de Diciembre del 2011, disponible en la web http://www.Adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf

En la siguiente tabla se puede observar la comparación entre metodologías ágiles y los métodos clásicos.

TABLA N° 1. 1. Bases para métodos ágiles y clásicos

AREA	METODOLOGIA AGIL	METODOS CLASICOS
Desarrolladores	Colaborativos, unidos, ágiles y entendidos	Orientados al plan con una mezcla de habilidades
Clientes	Son representativos y se les entrega poder	Mezcla de niveles de aptitud
Confianza	Conocimiento tácito interpersonal	Conocimiento explícito documentado
Requerimientos	En gran parte emergentes y con rápidos cambios	Conocibles tempranamente y bastante estables
Arquitectura	Diseñada para los requerimientos actuales	Diseñada para los actuales y los del futuro próximo
Refactorización	Económica	Costosa
Tamaño	Productos y equipo pequeños	Productos y equipos más grandes
Premium	Valor rápido	Alta seguridad

FUENTE. http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf

1.9.2. Metodologías ágiles para el desarrollo de software móvil

Conforme a AGILEDICTEMASANV en su artículo Metodologías Ágiles menciona que “Las metodologías ágiles poseen ciertas propiedades que las hacen totalmente aplicables al dominio del software en los móviles. Se identifican los métodos ágiles como la solución potencial para el desarrollo de software móvil.

Se apoya en las bases haciendo un análisis comparativo para probar la capacidad de los métodos ágiles sobre el desarrollo de software para móviles.” 14 de Diciembre del 2011, disponible en la web http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf

En el siguiente grafico se muestran las características ideales y su motivación en metodologías ágiles para desarrollar aplicaciones.

TABLA N° 1. 2. Características básicas ágiles y rasgos en el desarrollo de software móvil

Características ágiles	Motivación lógica	En el caso de desarrollo para plataformas móviles
Alta volatilidad del entorno	Debido a la alta frecuencia en el cambio que sufren los requerimientos, menos necesidad de diseño y planificación inicial y mayor necesidad de desarrollos incrementales e iterativos	Alta incertidumbre, entornos dinámicos, cientos de nuevos terminales cada año
Equipos de desarrollo pequeños	Capacidad de reacción más rápida, trabajo basado a la compartición de la información, menos documentación	La mayor parte de los proyectos de desarrollo de Sw para plataformas móviles se lleva a cabo en microempresas PyME
Cliente identificable	Desaparecen los malentendidos	Potencialmente, hay un número ilimitado de usuarios finales, pero los clientes son fáciles de identificar
Entornos de desarrollo orientados a objetos	Mayoría de las herramientas de desarrollo ágil existen bajo plataformas orientadas a objetos	Por ejemplo, Java y C++ se usan, algunos problemas en herramientas como refactorizaciones o primeros tests
SW. critico no asegurado	Los fallos no causan gran impacto, como la pérdida de vidas. Se pueden buscar mayor agilidad en el desarrollo	La mayoría de Sw. Es para entretenimiento. Los terminales no son fiables
SW. A nivel de aplicación	Sistemas embebidos grandes requieren comunicación exhaustiva y mecanismos de verificación	Mientras los sistemas móviles son complejos y altamente dependientes, las aplicaciones son muy autónomas
Sistemas pequeños	Menos necesidad de diseño inicial	Las aplicaciones, aunque variables en tamaño, no suelen superar las 10.000 líneas de código
Ciclos de desarrollo cortos	Propósito de realimentación rápida	Periodos de desarrollo de 1 a 6 meses

FUENTE. http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf

Para desarrollar software ya sea para dispositivos móviles, para escritorio o web se debe seleccionar una metodología adecuada y acorde a las necesidades de la aplicación y requerimientos del usuario.

De la misma manera en la dirección antes mencionada indica las características que debe tener un dispositivo móvil para el desarrollo de aplicaciones:

1.9.2.1. Características y requerimientos del entorno móvil

El desarrollo de aplicaciones móviles es diferente del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque el software móvil tiene que satisfacer una serie de requerimientos y condiciones que lo hace más complejo:

- **Canal radio:** Consideraciones como disponibilidad, desconexiones, variabilidad del ancho de banda, riesgos de seguridad se deben tenerse en cuenta en este entorno de comunicaciones móviles.
- **Movilidad:** Consideraciones como migración de direcciones, alta latencia debido a cambio de estación base o gestión de la información dependiente de localización aquí se pueden implementar un sinnúmero de aplicaciones.
- **Portabilidad:** La portabilidad de los dispositivos implica una serie de limitaciones físicas directamente relacionadas con el factor forma de los mismos, como el tamaño de las pantallas o del teclado, limitando también el número de teclas y su disposición.
- **Fragmentación de la industria:** La existencia de una considerable variedad de estándares, protocolos y tecnologías de red, complejidad al escenario del desarrollo móvil.
- **Capacidades limitadas de los terminales:** Factores como la baja potencia de cálculo o gráfica, riesgos en la integridad de datos, interfaces de usuario poco funcionales en muchos aspectos, baja capacidad de almacenamiento, la duración de las baterías, etc.
- **Time-to-market:** Los requisitos que se imponen en términos tiempo de lanzamiento son muy estrictos y añaden poca dificultad en la gestión de los procesos de desarrollo.

- **Diseño:** El diseño multitarea y la interrupción de tareas es clave para el éxito de las aplicaciones de escritorio; pero la oportunidad y frecuencia de éstas debido al entorno móvil, se complica más debido a las limitaciones de los dispositivos.
- **Usabilidad:** Las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos.

Puedo manifestar que el diseño de sistemas de software móvil es, por tanto, bastante más complejo que el tradicional (de escritorio), visto en los otros proyectos de desarrollo, forzando a los investigadores a reconsiderar el uso de las metodologías actuales de desarrollo de software. Como se ha visto, el uso de metodologías ágiles es el medio más apropiado para el desarrollo de tecnología en móviles, aunque las características especiales de los terminales y de las redes de telefonía móvil demandan algunos ajustes sobre las actuales metodologías ágiles.

CAPÍTULO II

DESCRIPCIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

2.1. Introducción

El propósito de la presente investigación es presentar la organización e interpretación de los resultados obtenidos de la investigación sobre EL ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, el desarrollo del proyecto se realizó a través de la INVESTIGACION DE CAMPO ya que se apoya en la información que se obtienen de los instrumentos de investigación como es la encuesta que se realizó a docentes y alumnos de la Carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.

Para una representación clara de la información recopilada se hará uso del programa Excel para la representación de tablas, gráficos circulares y finalizar con el análisis e interpretación de resultados.

2.2. Caracterización de la Universidad Técnica de Cotopaxi

De acuerdo al artículo publicado en la página oficial de la universidad se encuentra que “La Universidad Técnica de Cotopaxi es una Institución de Educación Superior, pública autónoma laica y gratuita, creada mediante ley promulgada en el Registro Oficial N° 618 del 24 de enero de 1995, fecha desde la que forma parte del Sistema Nacional de Educación Superior Ecuatoriano. Se rige por la Constitución Política del Estado, La Ley de la Educación Superior y otras leyes conexas.

Es una institución Universitaria sin fines de lucro que orienta su trabajo hacia los sectores urbanos, marginales y campesinos; que busca la verdad y la afirmación de la identidad nacional y que asume con responsabilidad el aseguramiento de la libertad en la producción y difusión de los conocimientos y del pensamiento democrático y progresista para el desarrollo de la conciencia antiimperialista del pueblo.” 15 de Enero del 2012, disponible en la página web www.utc.edu.ec/

2.2.1. Reseña Histórica de la Universidad Técnica de Cotopaxi

De acuerdo al artículo publicado en la página oficial de la universidad se encuentra que “La Universidad Técnica de Cotopaxi surgió en 1992 con la extensión de la Universidad Técnica del Norte y que fruto de varias movilizaciones de la gente de Cotopaxi se creó el 24 de enero de 1995.

El local de la UNE-C fue la primera morada administrativa; luego las instalaciones del colegio Luis Fernando Ruiz que acogió a los Universitarios; posteriormente el Instituto Agropecuario Simón Rodríguez, fue el escenario de las actividades académicas: para finalmente instalarse en casa propia, en un edificio a medio construir y que estaba destinado a ser Centro de Rehabilitación Social.

En la actualidad son cinco hectáreas las que forman el campus y 82 las del Centro Experimentación, Investigación y Producción Salache. Se ha definido con claridad la postura institucional ante los dilemas internacionales y locales.

Es una Institución antiimperialista porque rechaza frontalmente la agresión globalizadora de corte neoliberal que privilegia la acción fracasada economía de libre mercado.

En los 17 años de vida institucional la madurez ha logrado ese crisol emancipador y de lucha en bien de la colectividad, en especial de la más apartada y urgida en atender sus necesidades. El nuevo reto institucional cuenta con el compromiso constante de sus autoridades hacia la calidad y excelencia educativa.” 15 de Enero del 2012, disponible en la página web <http://www.utc.edu.ec/es-es/lautc/historia.aspx>

2.2.2. Misión Institucional

De acuerdo al artículo publicado en la página oficial de la universidad se encuentra que “Es una universidad pública, laica y gratuita, con plena autonomía, desarrolla una educación liberadora, para la transformación social, satisface las demandas de formación y superación profesional, en el avance científico-tecnológico de la sociedad, en el desarrollo cultural, universal y ancestral de la población ecuatoriana.

Generadora de ciencia, investigación y tecnología con sentido: humanista, de equidad, de conservación ambiental, de compromiso social y de reconocimiento de la interculturalidad; para ello, desarrolla la actividad académica de calidad, potencia la investigación científica, se vincula fuertemente con la colectividad y lidera una gestión participativa y transparente, con niveles de eficiencia, eficacia y efectividad, para lograr una sociedad justa y equitativa.

2.2.3. Visión

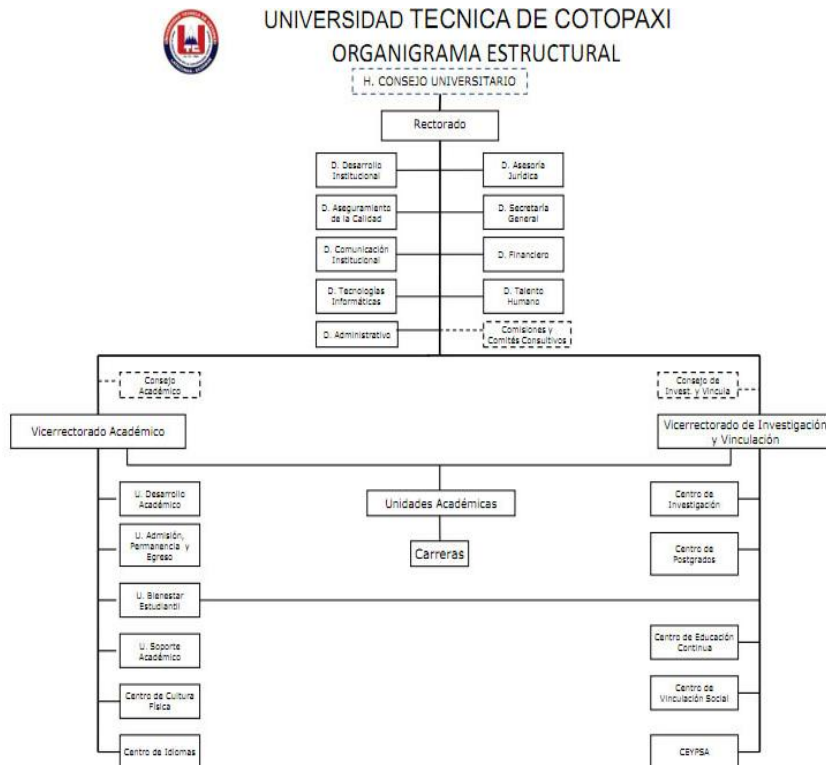
Universidad líder a nivel nacional en la formación integral de profesionales, con una planta docente de excelencia a tiempo completo, que genere proyectos investigativos, comunitarios y de prestación de servicios, que aporten al desarrollo local, regional en un marco de alianzas estratégicas nacionales e internacionales.

Difunde el arte, la cultura y el deporte, dotada de una infraestructura adecuada que permite el cumplimiento de actividades académicas, científicas, tecnológicas, recreativas y culturales, fundamentadas en la práctica axiológica y de compromiso social, con la participación activa del personal administrativo profesional y capacitado.” 15 de Enero del 2012, disponible en la página web <http://www.utc.edu.ec/es-es/lautc/misionvision.aspx>

2.3. Organigrama estructural

A continuación se muestra el organigrama estructural de la Universidad Técnica de Cotopaxi, esta comprende desde honorable consejo universitario, rectorado, vicerrectorado académico, vicerrectorado de investigación y vinculación, unidades académicas, y las carreras de estudio.

GRAFICO N° 2. 1. Organigrama Estructural U.T.C.



FUENTE: <http://www.utc.edu.ec/es-es/lautc/organigrama.aspx>

2.4. Descripción de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas

La Unidad Académica de Ciencias de la Ingeniería y Aplicadas de las ingenierías técnicas en general actualizadas y acordes a al pensum que el alumno necesita para sus conocimientos en cada uno de los perfiles profesionales.

2.4.1. Misión

De acuerdo al artículo publicado en la página oficial de la universidad se encuentra que la Misión de la Unidad CIYA es “Formar profesionales creativos, críticos y humanistas que utilizan el conocimiento científico y técnico, mediante la promoción y ejecución de actividades de investigación y aplicaciones tecnológicas para contribuir en la solución de los problemas de la sociedad.

2.4.2. Visión

Una Unidad Académica con un alto nivel científico, investigativo, técnico y profundamente humanista, generadora de tecnologías, con trabajos inter y multidisciplinario, que se concretan en proyectos investigativos, productivos, comunitarios y de prestación de servicios, a través de convenios con instituciones públicas y privadas, locales, nacionales e internacionales con una administración democrática, horizontal, vinculada con la sociedad. ”15 de Enero del 2012, disponible en la página web <http://www.utc.edu.ec/utc3/es-es/uacademicas/ciya.aspx>

2.5. Descripción de la Carrera de Ingeniería en Informática y Sistemas Computacionales

De acuerdo al artículo publicado en la página oficial de la universidad se encuentra que “La carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi, fue creada en el año de 1997 como respuesta a las demandas del mercado.

Su pensum y programas de estudio se han venido actualizando periódicamente para mantenerlo al ritmo de los cambios de la disciplina y de la tecnología que se usa en la profesión.

La UTC propone la Carrera de Ingeniería en Informática y Sistemas Computacionales para preparar profesionales capaces de cumplir las demandas de los usuarios informáticos en las organizaciones, con calidad, técnica, personal, moral y con profundo sentido social, para no solo ocupar puestos de trabajo sino ser capaces de generarlos en miras al desarrollo social del país.

Así mismo complementa la gama de carreras y especialidades que ofrece con ésta de gran impacto social y económico en el momento actual, además de ser capaz de autoabastecerse en la demanda de cursos en el área informática para otras carreras y soluciones informáticas que las tendencias de la institución requieren.

2.5.1. Misión

La Carrera de Ingeniería en Informática y Sistemas Computacionales se encamina a formar Ingenieros con una misión integradora del hombre, la sociedad y de su labor profesional, entendida ésta como servicio y generadora de cambios sociales, por medio de la síntesis de los saberes humanísticos, tecnológicos y científicos.

Fomentar el perfeccionamiento integral de todos los miembros de la comunidad académica, con una atención personalizada, mediante un ejercicio académico creativo, riguroso e interdisciplinario.

Promueve la realización de investigaciones o proyectos docentes interdisciplinarios en las áreas de la ingeniería que sean de su competencia, y fomenta la realización del trabajo vivido como servicio y medio para construir una sociedad justa, pacífica y solidaria. Esto se logra en un ambiente de libertad y responsabilidad personales, y de reconocimiento del hombre como ser trascendente, conforme a una visión real de la vida.

2.5.2. Visión

El Programa de Ingeniería en Informática y Sistemas Computacionales espera ganarse el reconocimiento tanto local como nacional en el área de su competencia mediante la conjugación de tres elementos esenciales: investigación, academia y vinculación a la sociedad a través de:

- Desarrollo de procesos formativos que logren que los profesionales sean altamente competitivos e influyentes en los sectores que produzcan o utilicen tecnologías informáticas.
- Perfeccionamiento de los procesos permanentes de autoevaluación del programa.
- La continúa interacción con pares académicos nacionales.
- La selección de profesores de la más alta calidad intelectual y humana.
- La creación de líneas de investigación que apunten al trabajo en tecnologías de punta y al planteamiento de alternativas de solución a problemas en el entorno.
- La creación de programas de Postgrado en diferentes áreas pertinentes al ámbito de la ingeniería de software y telecomunicaciones.
- El establecimiento de alianzas o convenios estratégicos con empresas o industrias locales y nacionales.

2.5.3. Perfil Profesional

El Ingeniero en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi, es un profesional con un dominio de la teoría y tecnología de punta tanto en hardware como software, a través de:

- Planificar, analizar, diseñar, seleccionar, construir, operar, mantener, integrar, evaluar, optimizar y auditar sistemas de información, aplicados en las áreas administrativas, técnicas, científicas y sociales.
- Analizar, diseñar e implementar Sistemas Informáticos.

- Proveer tecnologías de mejoramiento de procesos organizacionales.
- Aplicar y construir metodologías y planes de acción para enfrentar problemas informáticos a corto, mediano y largo plazo.
- Diseñar, implementar y administrar redes de computadoras y sistemas digitales.
- Aplicar software utilitario y paquetes informáticos.
- Asesorar procesos de evaluación y control de plataformas de Hardware y Software.
- Incorporar los avances de la tecnología de la informática en la investigación científica.
- Analizar, construir y administrar bases de datos en distintas plataformas.” 15 de Enero del 2012, disponible en la página web <http://www.utc.edu.ec/utc3/es-es/uacademicas/ciya/sistemas.aspx>

Pienso que la Universidad Técnica de Cotopaxi desarrolla una actividad docente con niveles adecuados de calidad, brindando una oferta educativa en pregrado y posgrado, formando profesionales analíticos, críticos, investigadores, humanistas capaces de generar ciencia y tecnología. Asimismo, realiza una actividad científico-investigativa que le permite brindar aportes en la solución de los problemas más importantes, se distingue de otras instituciones de educación superior al ser una universidad alternativa vinculada fuertemente con el pueblo en todas sus actividades de ahí nace el eslogan “Por la Vinculación de la Universidad con el Pueblo”.

2.6. Unidad de Estudio

2.6.1. Población

Para el proceso de estudio se ha tomado en cuenta la siguiente población que faciliten la información necesaria y que sea de aporte en el proyecto investigativo.

TABLA N° 2. 1. Población a investigar

INVOLUCRADOS	CANTIDAD
Estudiantes de la carrera de informática y sistemas computacionales	284
Docentes de la carrera de informática y sistemas computacionales	25
TOTAL	309

Fuente: Sra. Paola Segovia (Secretaria de la Unidad de Ciencias de la Ingeniería y Aplicadas)

2.6.2. Muestra

Debido a que la población fue de 309 personas, se utilizara la siguiente formula para el calculo de la muestra que dara el total a investigar.

$$n = \frac{N \cdot O^2 Z^2}{(N - 1)E^2 + O^2 Z^2}$$

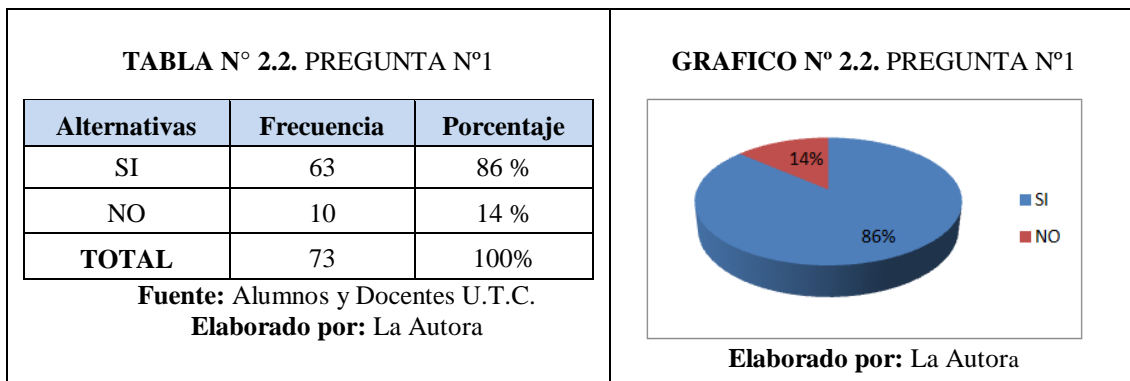
$$n = \frac{309 \cdot (0.25)^2 1.96^2}{(309 - 1)(0.05)^2 + (0.25)(1.96)^2}$$

$$n = 73 \Rightarrow \text{valor total de la muestra a investigar}$$

2.7. Análisis e interpretación de resultados de las encuestas realizadas a docentes y alumnos de la Carrera de Sistemas

PREGUNTA N°1

¿Cree usted que el desarrollo de aplicaciones móviles es lo más novedoso en la actualidad?



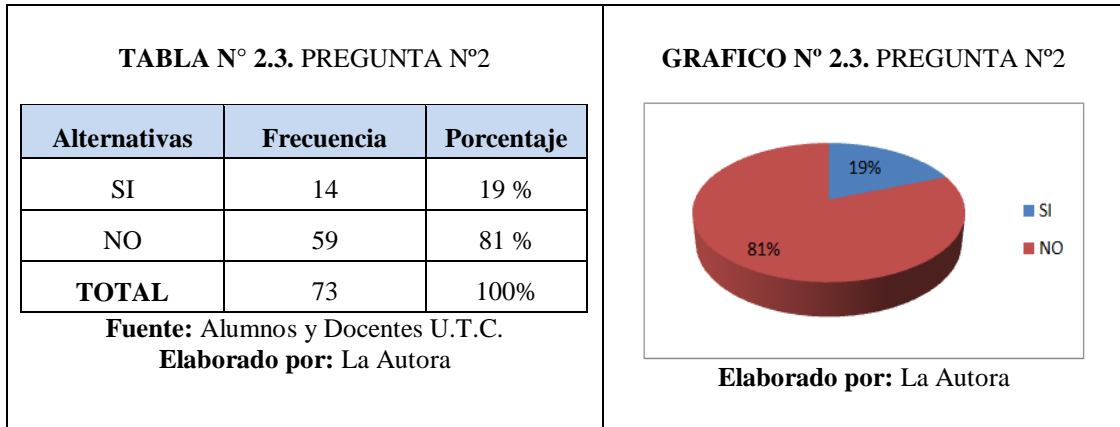
INTERPRETACIÓN.

De las encuestas realizadas a estudiantes y docentes, el 86% contestaron que sí, y el 14% contestaron que no, por lo que es notorio que el sí prevalece en esta pregunta.

Mediante la pregunta planteada se evidencia que el SI tiene mayor porcentaje, por lo que se deduce, que hoy en día la movilidad es la tecnología actual y dentro de la misma el desarrollo de software para dispositivos y que los encuestados tienen un gran interés por conocer.

PREGUNTA N°2

¿Ha incursionado alguna vez en el desarrollo de aplicaciones móviles?



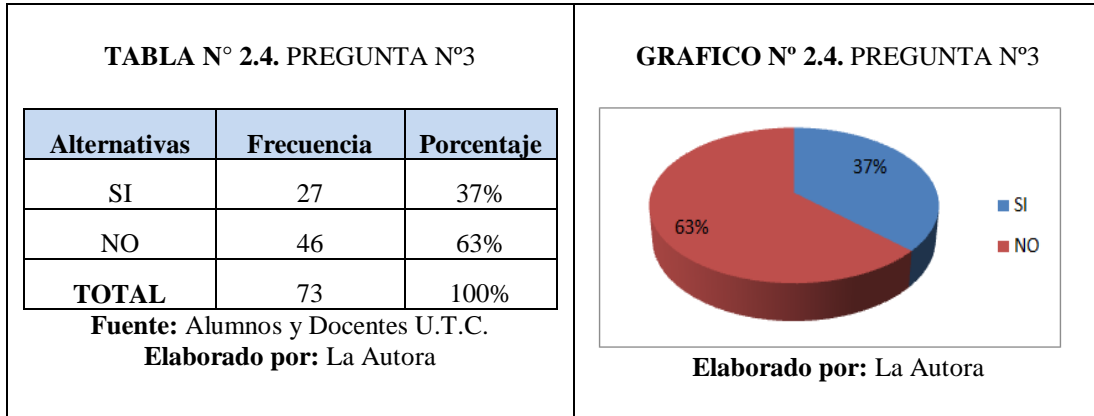
INTERPRETACIÓN.

En la presente pregunta del 100% de encuestados entre estudiantes y docentes, 14 equivalente al 19% dicen que sí, mientras que 59 equivalente al 81% dicen que no, por lo que se puede evidenciar que el mayor porcentaje recae en el NO.

Como se puede observar existe gran desconocimiento en la mayoría de encuestados ya que no han incursionado en el desarrollo de aplicaciones móviles, por tal razón es la necesidad inmediata de propiciar al desarrollo de este tipo de aplicaciones, ya sea a través de capacitación sobre este tipo de tecnología o con el desarrollo de este proyecto en mención para en un futuro realizar exposiciones del mismo.

PREGUNTA N°3

¿Ha descargado e instalado aplicaciones para su teléfono celular que tenga sistema operativo Android?



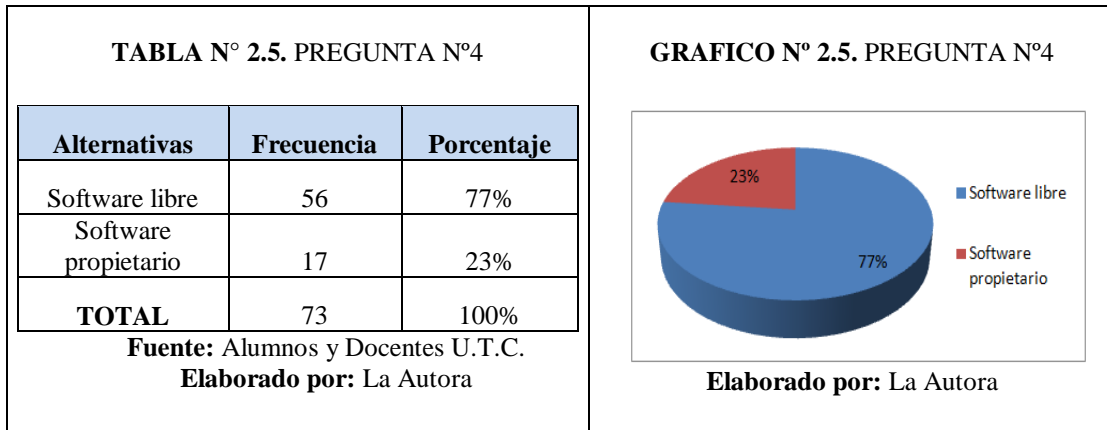
INTERPRETACIÓN.

Del total de encuestados, en la presente pregunta el 37% equivalente a 27 encuestados se inclinan por el sí, y el 63% equivalente a los 46 encuestados se inclinan por el no, lo que es claro que la mayoría en el resultado de la encuesta se inclinan por el NO.

Al verificar que la mayor parte de encuestados dicen que **NO han descargado, ni instalado** aplicaciones para sus celulares siendo estas **fáciles de realizarlas**, el presente proyecto de tesis es un referente para incentivar a las descargas de este tipo de aplicaciones, y en especial para aquellos dispositivos que posean un Sistema Operativo Android.

PREGUNTA N°4

¿Qué tipos de licencias considera usted utilizar para desarrollar aplicaciones para dispositivos móviles?



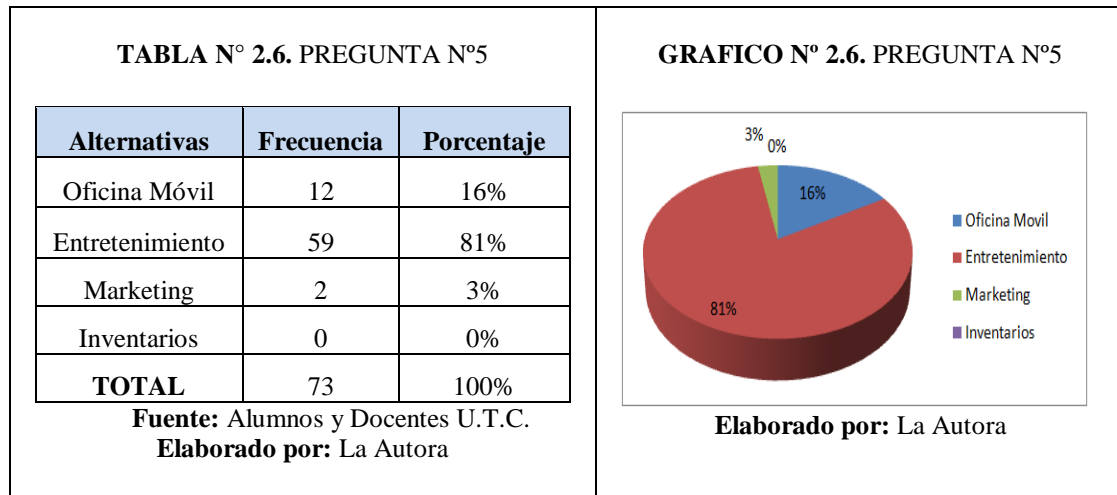
INTERPRETACIÓN.

Del 100% de encuestados, el 77% equivalente a 56 personas consideran mejor utilizar software libre en el desarrollo de aplicaciones móviles, el 23% restante equivalente a 17 personas se inclinan por software propietario, es notorio que el uso de Software Libre llama la atención en docentes y estudiantes.

Una vez verificado que la mayor parte de encuestados se inclinan por el uso de Software Libre, en el presente proyecto se enfoca en el análisis y estudio de herramientas de distribución libre, para aprovechar las bondades del mismo, sin dejar atrás que el software propietario tiene sus ventajas pero también limitaciones al momento de usar sus herramientas, y que por lo cual se trata de **incentivar** el desarrollo de aplicaciones móviles a estudiantes y docentes dejando que sean ellos mismos los que decidan el uso de herramientas ya sean **software libre o software propietario**.

PREGUNTA N° 5

¿Para qué fin ha utilizado las aplicaciones de su celular?



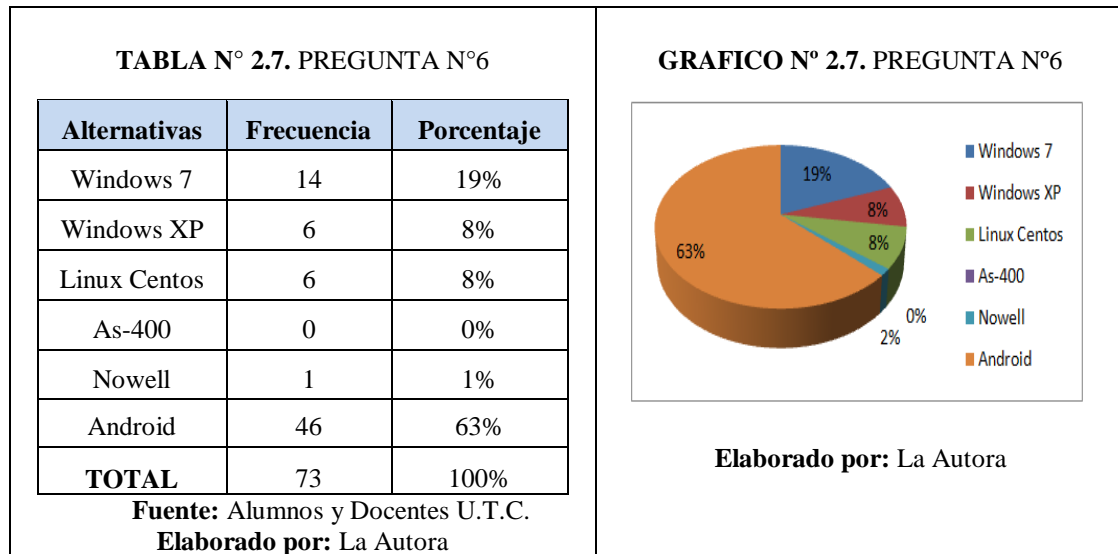
INTERPRETACIÓN.

En la presente pregunta existen varias apreciaciones en los resultados de la encuesta recayendo que el 81% (59 encuestados) dicen utilizar las aplicaciones de su celular para entretenimiento, el 16% (12 encuestados) para usar aplicaciones de oficina móvil, y un porcentaje reducido el 3% (2 encuestados) para uso de aplicaciones de marketing y un 0% de inventarios, entonces evidenciamos que la mayor parte usan aplicaciones de sus celulares para entretenimiento.

Una gran parte de encuestados usan aplicaciones de entretenimiento en sus dispositivos móviles (celular), es primordial fomentar el desarrollo y uso adecuado de dichas aplicaciones, **incentivando** a estudiantes y docentes al desarrollo de las mismas para así fortalecer su aprendizaje en su formación profesional y/o académica.

PREGUNTA N°6

Cuál considera usted que es un sistema operativo para dispositivos móviles?



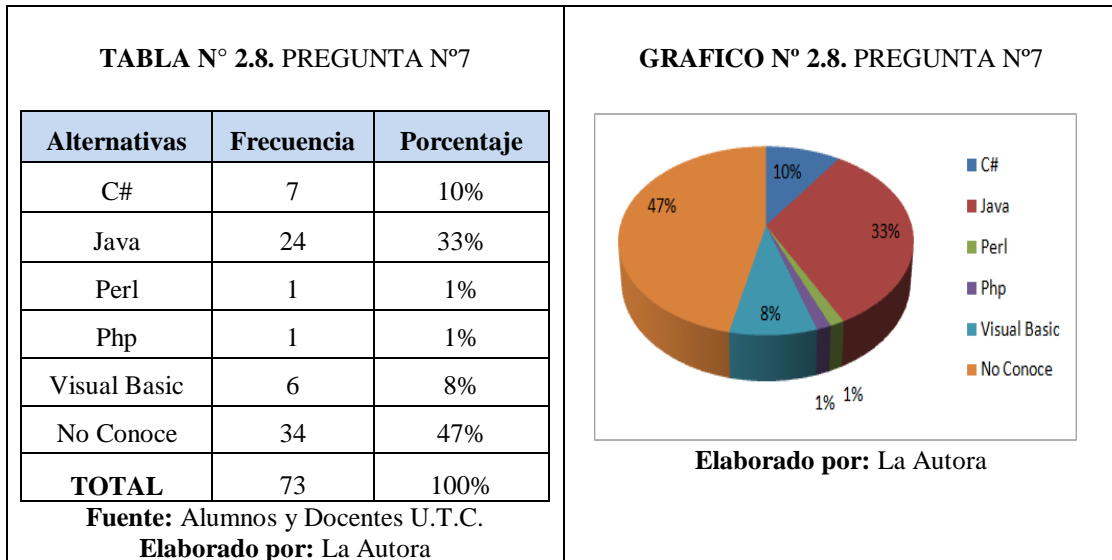
INTERPRETACIÓN.

De la muestra seleccionada para la encuesta el 63% conocen a Android como un sistema operativo para dispositivos móviles, el 19% dicen que Windows 7, un 8% dicen que Windows XP, un 8% dicen que Linux Centos un 0% por As-400, y un 1% dicen que Novell, lo que se deduce que más del 50%de encuestados conocen con exactitud sobre el Sistema Operativo Android.

Es evidente que la mayor parte de usuarios de celulares conocen sobre un Sistema Operativo, y en especial sobre Android que es instalado hoy en día en diferentes dispositivos móviles por ser gratuito. Entonces se considera factible en nuestro proyecto la utilización y estudio de este Sistema Operativo y sobre este desarrollar las aplicaciones que hacen de un dispositivo móvil más versátil y eficiente en la ejecución de sus aplicaciones desarrolladas con herramientas de distribución libre.

PREGUNTA N°7

¿Ha desarrollado aplicaciones para dispositivos móviles. Especifique en que lenguaje de programación?



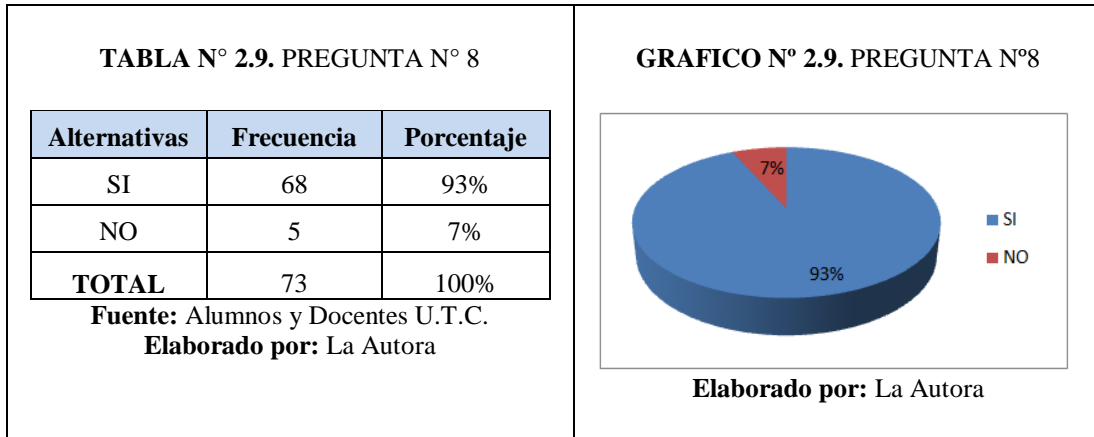
INTERPRETACIÓN.

Con relación a la presente pregunta, se trata de analizar si han utilizado algún lenguaje de programación en el desarrollo de aplicaciones móviles, y como se puede observar un gran porcentaje como es el 33% conocen Java y en un 47% no lo conocen, cabe destacar que un 10% dicen C# y un 8% dicen conocer Visual Basic y un 1% a PhpyPerl.

Interpretando los resultados obtenidos, el estudio de herramientas libres y como lenguaje de programación en el presente proyecto nos enfocamos en Java y por **desconocimiento** de la gran mayoría, el análisis y estudio de java como lenguaje de programación es esencial para el desarrollo de aplicaciones móviles, para así fortalecer los conocimientos de estudiantes y docentes en la aplicación de este lenguaje que es netamente orientado a objetos.

PREGUNTA N°8

¿Ayudaría en su aprendizaje conocer el proceso de cómo se desarrollan aplicaciones para dispositivos móviles (celular)?



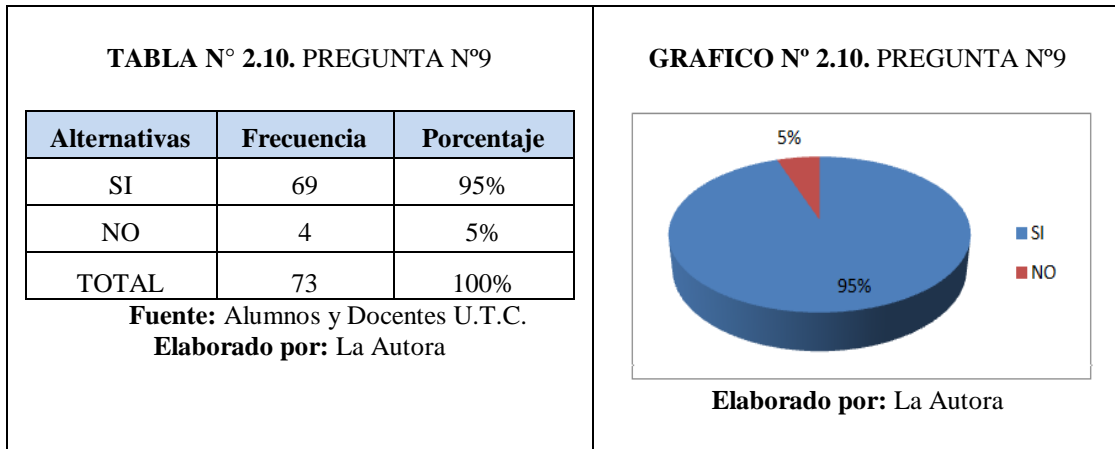
INTERPRETACIÓN.

De la muestra obtenida de la población el 93% coinciden que les gustaría seguir un proceso para el desarrollo de aplicaciones móviles ya que ayudaría en el aprendizaje de su carrera estudiantil y como docente profesional, mientras que el 7% de encuestados dicen que no ayudaría en su proceso de aprendizaje.

Cabe destacar que la mayor parte de encuestados están de acuerdo que si ayudaría en su aprendizaje ya que el desarrollo de software requiere de un adecuado y riguroso proceso, y para ello el presente proyecto propone seguir metodologías de desarrollo con lo que al final se logre obtener un producto de calidad.

PREGUNTA N°9

¿Le gustaría conocer nuevas herramientas para el desarrollo de aplicaciones móviles sobre un sistema operativo Android?



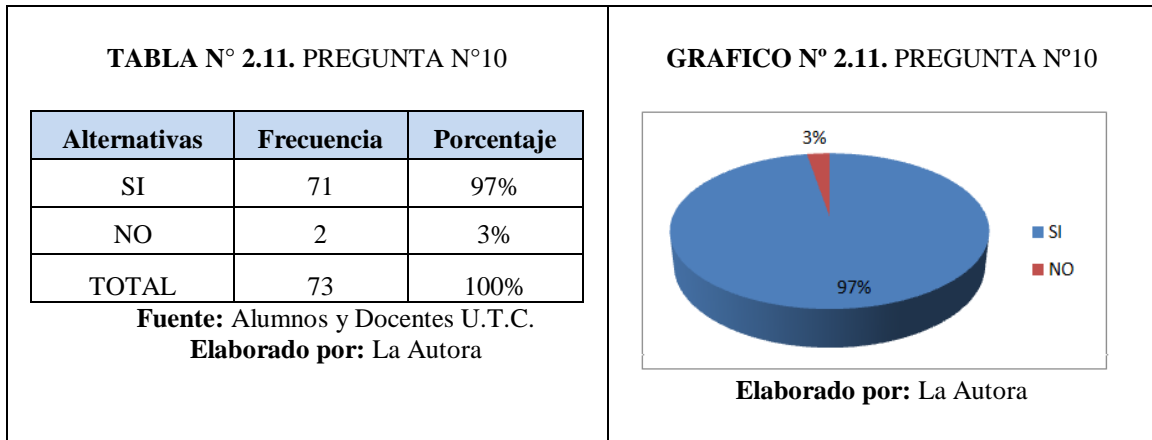
INTERPRETACIÓN.

De las encuestas realizadas entre docentes y estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi, el 95% equivalente a 69 encuestados dicen que si les gustaría conocer de nuevas herramientas para desarrollo de aplicaciones, mientras que el 5% equivalente a 4 encuestados dicen que no les gustaría conocer.

Entonces evidenciando este referente que la mayor parte de encuestados si les gustaría conocer de nuevas herramientas para el desarrollo de aplicaciones móviles, se hace factible el analizar y estudiar dichas herramientas de distribución libre, y que se ejecuten sobre un Sistema Operativo Android.

PREGUNTA N°10

¿Le gustaría crear o desarrollar sus propias aplicaciones (software) para su celular?



INTERPRETACIÓN.

Ya que un 97% equivalente a 71 encuestados están de acuerdo en que les gustaría desarrollar sus propias aplicaciones para su teléfono celular, mientras que el 3% equivalente a 2 encuestados no están de acuerdo en desarrollar sus propias aplicaciones.

Considerando estos resultados la mayor parte de encuestados muestran un gran interés por conocer y aprender nuevas tecnologías y más aun si se trata de desarrollar aplicaciones móviles con su propia autoría, cabe resaltar que el desarrollo de aplicaciones móviles sería un incentivo para estudiantes y docentes de la carrera.

2.8. Verificación de la hipótesis

La hipótesis planteada en el proyecto de tesis decía: “El análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles, permitirá la construcción y uso adecuado de una aplicación para entretenimiento en usuarios de celulares de la Carrera de Ingeniería en Informática y Sistemas Computacionales.”

DECISIÓN.

De los resultados obtenidos de las encuestas aplicadas a los docentes y estudiantes de la Carrera de Ingeniería en Informática y Sistemas Computacionales se comprobó mediante las preguntas N°9 y N°10 que el 95 y 97% de los encuestados primero desearían conocer herramientas de desarrollo de aplicaciones para su celular con sistema operativo Android, de la misma manera les gustaría desarrollar sus propias aplicaciones para su celular, es por esta razón que la autora propone el análisis y estudio de herramientas libres para el desarrollo de aplicaciones de entretenimiento en celulares con sistema operativo Android.

Se deduce que con el análisis y estudio de herramientas libres en teléfonos celulares con sistema operativo android, ayudaría de gran manera a reducir el desconocimiento que existe por parte de los encuestados en la existencia de nuevas herramientas para desarrollar aplicaciones móviles, su función y como se las crea, sabiendo que en esta área de la informática día a día la tecnología avanza rápidamente y nosotros como estudiantes o docentes debemos estar siempre en la investigación de nuevas tendencias tecnológicas y como hoy en día es lo esencial la tecnología móvil incursionar y ser participes en el desarrollo de este tipo de aplicaciones y ser autores de un trabajo investigativo.

2.9. Metodología para el desarrollo de aplicaciones móviles

2.9.1. Metodología XP (Programación Extrema)

Según DAPENA, Jose, en su obra Desarrollo en Comunidad con eXtreme Programming manifiesta que “La programación extrema o eXtreme Programming (XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

2.9.1.1. Descripción de Extreme programming

Extreme Programming (Programación Extrema), es una metodología ágil de desarrollo de productos de software, que se focaliza en la satisfacción del cliente, manteniendo como doctrina “el entregar el software que el cliente necesita cuando lo necesita”.

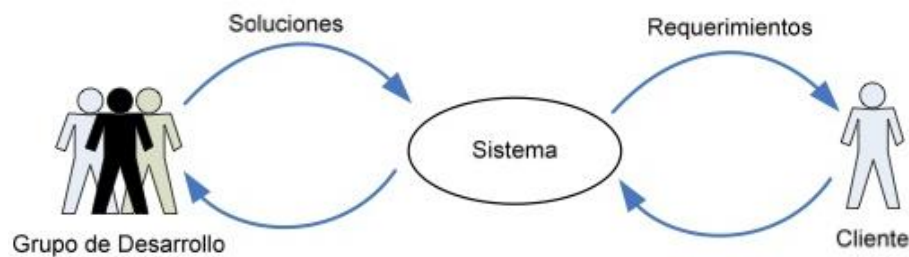
El trabajo en conjunto es uno de los pilares de Extreme Programming, donde el cliente forma parte del equipo de desarrollo, junto con los programadores, comprometiéndose a obtener un producto de calidad, disminuyendo de forma notable el tiempo invertido en las pruebas de aceptación del sistema.

El desarrollo de software usando XP(Extreme Programming), es fortalecido con cuatro aspectos importantes que son:

- Comunicación
- Simplicidad
- Valor
- Retroalimentación

XP es una metodología adecuada para equipos pequeños de desarrollo, y no requiere de personas expertas en un área en específico, dirigida a delimitar los riesgos del desarrollo, debido a que el sistema a desarrollar está de acuerdo a las necesidades cambiantes de un cliente.

GRAFICO N° 2.12. Desarrollo de un Sistema usando XP



FUENTE: DAPENA, José. Desarrollo en Comunidad con eXtreme Programming. Italia

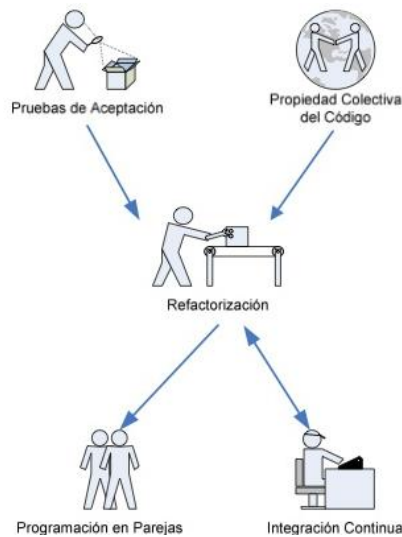
Una de las ventajas de XP es realizar entregas funcionales del sistema, por lo cual el grupo de desarrollo emplea prototipos evolutivos, y mediante un test de funcionalidad el cliente acepta la entrega o plantea cambios. Es una metodología de acción antes que una metodología de control; a diferencia de las metodologías tradicionales, XP no obliga a los miembros del equipo a tener demasiados documentos, restricciones y actividades que a lo largo del proceso de desarrollo, muchas veces incrementan el tiempo del mismo.

2.9.1.2. Características de Xp

La principal característica es que incentiva la programación, de tal forma que crea un ambiente ideal para trabajar en equipos pequeños que van desde 2 o más integrantes por grupo de desarrollo, siendo las bases de la programación extrema: la Programación en Parejas, Propiedad Colectiva del Código, Pruebas de Aceptación, Refactorización e Integración Continua.

Estas bases permiten a los desarrolladores participar de una forma activa en el desarrollo, independientemente de quien programó y de cómo lo hizo.”2007, Págs.28-40

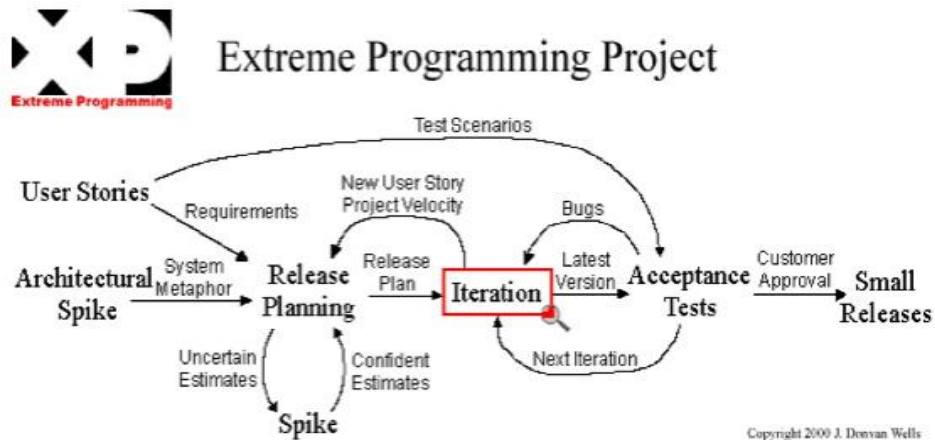
GRAFICO N° 2.13. Bases de eXtreme Programming



FUENTE: BECK, Kent. Extreme Programming Explained: Embrace Change. Primera Edición. Addison – Wesley. Estados Unidos de América. 2000.

2.9.1.3. Fases de la Metodología Xp

GRAFICO N° 2.14. Proceso de eXtreme Programming



Fuente: www.extremeprogramming.org

Las fases del ciclo de vida de un proceso XP son: Exploración, Planificación, Iteraciones, Producción, Mantenimiento y Muerte del proyecto.

2.9.1.3.1. Fase de Exploración

En la fase de exploración los desarrolladores deben familiarizarse con las herramientas que van a trabajar, además de aprender a trabajar entre sí. Durante la exploración, los programadores usan herramientas de desarrollo que luego serán manejadas en la producción del sistema, además exploran activamente las posibilidades para la arquitectura del sistema.

Los programadores también deben experimentar con los límites de la tecnología que están usando. Deben realizar simulaciones con cargas realísticas con el hardware y la red. Mientras el equipo de desarrollo está encargado de la tecnología, el cliente debe practicar la escritura de las historias.

2.9.1.3.2. Fase de Planificación

El propósito de esta fase es para los clientes y programadores estar de acuerdo en una fecha en la cual el conjunto de historias más pequeño y representativo va a ser realizado. La planificación, debe abarcar todas las historias a ser implementadas y repartidas en las iteraciones que se decidan efectuar.

La esencia de la planificación es para el equipo de desarrollo, estimar el tiempo en semanas que tarda en realizar cada historia de usuario, incluidas las pruebas. El cliente y los diseñadores deciden en conjunto qué historia es la más importante o tiene la prioridad más alta a ser completada.

2.9.1.3.3. Fase de Iteraciones

Esta fase requiere tener el plan de entrega definido, el cual contiene un conjunto de iteraciones y cada iteración tiene un conjunto de historias asignadas. Cada iteración origina un conjunto de casos de prueba funcionales para cada una de las historias fijadas para esta iteración.

2.9.1.3.4. Fase Puesta en Producción

En la fase de producción las impresiones del producto elaborado son recogidas, con el fin de poder mejorarlo en las próximas entregas, permitiendo así añadir características adicionales a la versión actual. XP propone realizar entregas funcionales del sistema, estas están listas para ser integradas al sistema una vez que hayan aprobado las pruebas unitarias, pero esto no significa que al momento de integrarlo con los distintos módulos, este funcione de forma correcta, por esta razón es necesario realizar pruebas al momento de integrar el sistema.

Durante esta etapa el equipo de trabajo baja el ritmo del desarrollo, sin que esto implique que el sistema deje de evolucionar, tomando en cuenta que mientras más sea la experiencia que tenga el grupo de trabajo en el sistema, mayor será su visión al momento de afrontar los cambios requeridos.

2.9.1.3.5. Fase de Mantenimiento

La Fase de mantenimiento es el estado normal de un proyecto de XP. Es necesario simultáneamente, producir la nueva funcionalidad, mantener el sistema existente corriendo e incorporar nuevas personas en el equipo.

2.9.1.3.6. Fase Muerte del Proyecto

Un proyecto llega a su final cuando el cliente no aporta con más historias de usuario para que el grupo de trabajo las implemente. Con esto puede darse por concluido el desarrollo del sistema, y es posible decir que el sistema ha alcanzado toda la funcionalidad que el cliente deseaba. Otra razón para que un proyecto concluya es la falta de recursos por parte del cliente, ocasionando que el proyecto sea abandonado.

CAPITULO III

DESARROLLO DE LA PROPUESTA

3.1. Tema

“ANÁLISIS Y ESTUDIO DE LAS HERRAMIENTAS LIBRES PARA EL DESARROLLO DE APLICACIONES MÓVILES EN TELÉFONOS CELULARES CON SISTEMA OPERATIVO ANDROID, Y LA CONSTRUCCIÓN DE UNA APLICACIÓN PARA ENTRETENIMIENTO DE USUARIOS EN LA CARRERA DE INFORMÁTICA Y SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI ”

3.2. Presentación

El análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles, permitirá la construcción y uso adecuado de una aplicación para entretenimiento en usuarios de celulares con sistema operativo android en la Carrera de Ingeniería en Informática y Sistemas Computacionales, el mismo que contribuirá en el desarrollo de nuevas tecnologías y aplicaciones móviles y más si en la actualidad es un tema importante en el mercado tecnológico el estudio y desarrollo de nuevos proyectos de software y aplicaciones móviles.

La aplicación de entretenimiento cuenta con la información concerniente a introducción a la aplicación, Ingreso de nuevos jugadores, Listado de jugadores, Editar jugadores, Ingreso de preguntas para el juego, Ingreso de opciones por cada pregunta, jugar aplicación, resumen de partida, información de desarrollo de la aplicación y por último las instrucciones de juego que debe tener en cuenta antes de empezar una partida y toda la documentación necesaria para crear un producto satisfactoriamente.

3.3. Justificación

El análisis y estudio de las herramientas libres para desarrollar aplicaciones móviles de entretenimiento son de gran importancia ya que se dará a conocer a usuarios de celulares que herramientas utilizar para desarrollar aplicaciones móviles y más si en la actualidad tiene gran importancia el software libre en el desarrollo de aplicaciones móviles.

En la actualidad en la carrera no se ha incursionado en el estudio de herramientas libres para desarrollar aplicaciones móviles, por lo que este tema es de gran aporte investigativo para los jóvenes estudiantes de la UTC que aun están en ciclos básicos de estudio en la Carrera.

Mediante los conocimientos adquiridos durante los ciclos de estudio, la información obtenida de fuentes bibliográficas como virtuales, facilitaran el proceso de desarrollo, la postulante a decidido analizar estudiar y construir una aplicación de entretenimiento para lo cual se utilizará herramientas libres como: JDK, Eclipse, SDK de android para java, Plugin de android para eclipse, lenguaje de programación java y xml son lenguajes excelentes para desarrollar aplicaciones móviles por su versatilidad y uso eficiente. Para la construcción de la base de datos se utilizara el gestor SQLITE por la seguridad y confiabilidad que brinda a la hora de manejar la información.

Así como también se contara con herramientas necesarias y recursos económicos suficientes que serán financiados por parte de la postulante ya que en el desarrollo de la aplicación se utilizara software libre con licencia GPL.

Para llevar a cabo la presente tesis se cuenta con el apoyo de dos profesionales en el ámbito técnico y metodológico, además con la colaboración de interesados e involucrados en el tema, quienes aportaran la información para la creación de la aplicación de entretenimiento y la postulante quien determinara los requisitos necesarios para el desarrollo de la aplicación.

3.4. Objetivos

Objetivo General

Realizar un análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles en teléfonos celulares con sistema operativo Android, para los usuarios de la Carrera de Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.

3.5. Factibilidad de aplicar la propuesta

Una vez planteada la propuesta de realizar un análisis y estudio de herramientas libres para desarrollar aplicaciones móviles en teléfonos celulares con sistema operativo android, se procedió a la recolección de información a docentes y estudiantes usuarios de celulares en la Carrera de Ingeniería en Informática y Sistemas Computacionales.

Luego de un análisis exhaustivo es factible analizar y estudiar herramientas libres para desarrollar aplicaciones móviles en teléfonos celulares con sistema operativo Android porque contribuirá con el desarrollo de nuevas tecnologías en nuestro medio y en especial en la Carrera de Ingeniería en Informática y Sistemas Computacionales lo cual es un aporte primordial en generaciones próximas a graduarse en la Universidad.

3.6. Desarrollo de la aplicación

3.6.1. Fase de Exploración

En la primera fase de XP se debe analizar la viabilidad del proyecto y estudiar su utilidad y si se tiene las capacidades necesarias para llevarlo adelante. El equipo de desarrollo debe familiarizarse con las herramientas y tecnologías que se utilizarán en el proyecto para conocer límites y rendimiento. Explorar las posibilidades de la arquitectura del sistema construyendo un prototipo.

Para el desarrollo de la presente aplicación se utilizaran herramientas actuales y sofisticadas por lo que se hará un previo estudio de funcionalidad de cada una. A continuación se describen las herramientas a utilizarse en el desarrollo de la aplicación.

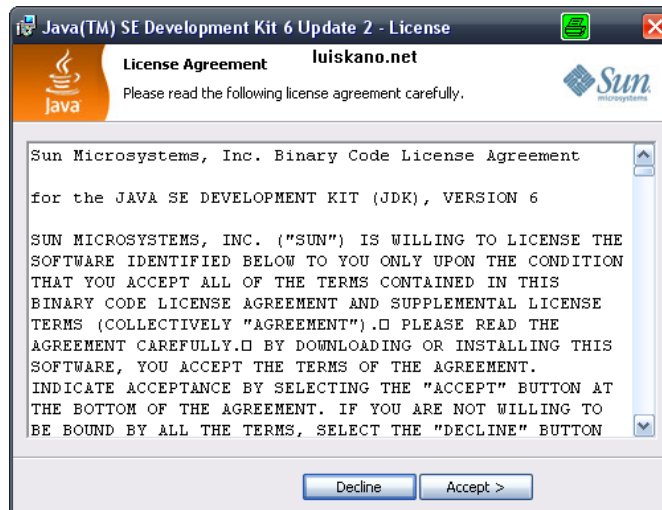
3.6.1.1. JDK(Java Development Kit)

“JDK es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una Computadora local o en una unidad de Red. Es la maquina virtual de java”

INSTALACIÓN Y CONFIGURACIÓN JDK

- 1) Ejecutar el archivo descargado y aparecerá una pantalla como esta, clic en **Accept**

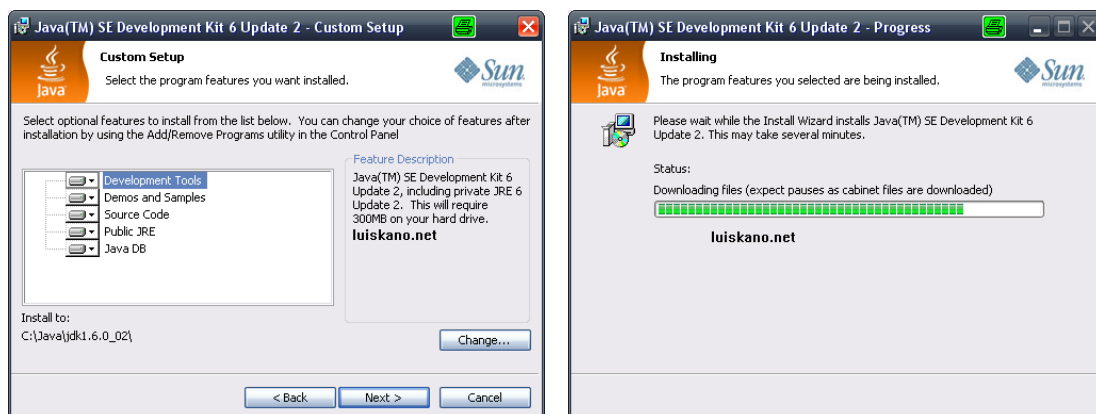
GRAFICO N° 3.1. Ejecutar Archivo JDK



REALIZADO POR: La Autora

- 2) Aparecerá una pantalla pidiendo donde instalar el jdk, [dejar la dirección por default], en **C:/Archivos de programas/**. Clic en **Next** y empezara la instalación de Java Development Tools.

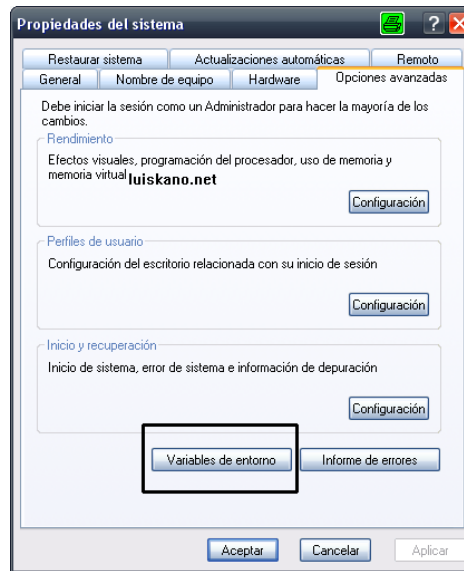
GRAFICO N° 3.2. Instalación JDK



REALIZADO POR: La Autora

- 3) Luego pedirá donde intalar el Java SE. [dejar la carpeta de instalación por Default], en **C:/Archivos de programas/**.
- 4) Click en **Next** y empezará la instalación de Java SE, clic en **Finish**:
- 5) Luego ir a **MI PC**, clic derecho **Propiedades->Opciones Avanzadas**. Clic en **Variable de entorno**.

GRAFICO N° 3.3. Configuración variable de entorno

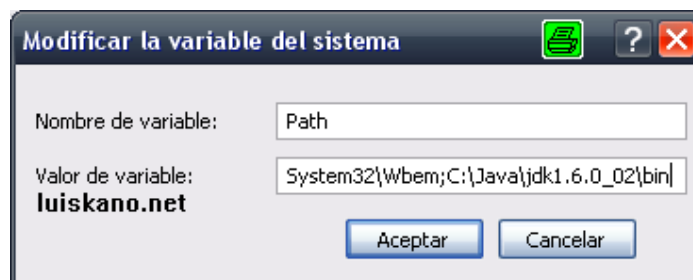


REALIZADO POR: La Autora

- 6) Buscar la variable **“Path”** y clic en **modificar**. En el campo **“valor de variable”** escribir la ubicación del compilador de JAVA, [que se dejo por default] **“;C: \Archivos de programas\.”**

Nota: Mirar que hay antes de ;c: esto separa el directorio de los otros valores ya ingresados

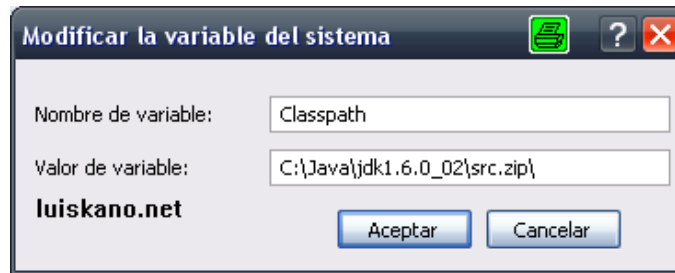
GRAFICO N° 3.4. Configuración variable path



REALIZADO POR: La Autora

- 7) Crear una variable llamada **Classpath**, ingresar la dirección de la variable. En este caso la dirección es “;C: \Archivos de programas\.” o si se creó la carpeta es ;C:\Java\jdk1.6.0_02\src.zip\ Para terminar clic en **Aceptar**.

GRAFICO N° 3.5. Configuración variable Classpath



REALIZADO POR: La Autora

- 8) Ir a **Inicio->Todos los programas-> Accesorios -> Símbolo del sistema**. Aparecerá una Consola DOS, escribir **java** y enter, luego **javac** para ver si el sistema encuentra el intérprete de JAVA.

GRAFICO N° 3.6. Consola DOS para actualizar intérprete de Java

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
luiskano.net

C:\Documents and Settings\Luis Kano>java
Usage: java [-options] class [args...]
       (to execute a class)
or java [-options] -jar jarfile [args...]
       (to execute a jar file)

where options include:
 -client          to select the "client" VM
 -server         to select the "server" VM
 -hotspot        is a synonym for the "client" VM [deprecated]
                 The default VM is client.

 -cp <class search path of directories and zip/jar files>
 -classpath <class search path of directories and zip/jar files>
              ; separated list of directories, JAR archives,
              and ZIP archives to search for class files.
 -D<name>=<value>
               set a system property
 -verbose[:class[:jini]]
               enable verbose output
 -version        print product version and exit
 -version:<value>
               require the specified version to run
 -showversion   print product version and continue
 -jre-restrict-search | -jre-no-restrict-search
               include/exclude user private JREs in the version search
 -? -help       print this help message
 -X            print help on non-standard options
 -ea[:<packagename>...[:<classname>]]
               enable assertions
 -da[:<packagename>...[:<classname>]]
               disable assertions
```

REALIZADO POR: La Autora

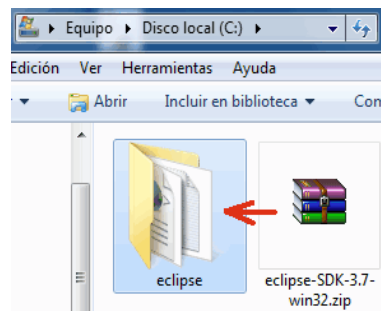
3.6.1.2. IDE ECLIPSE

“Eclipse es un IDE para el desarrollo de software de código abierto está escrito totalmente en Java permite crear aplicaciones empresariales, móviles, etc gracias a sus recursos disponibles”

INSTALACIÓN Y CONFIGURACIÓN DE ECLIPSE

- 1) Descomprimir el archivo descargado, pegar la carpeta eclipse en el disco C:\

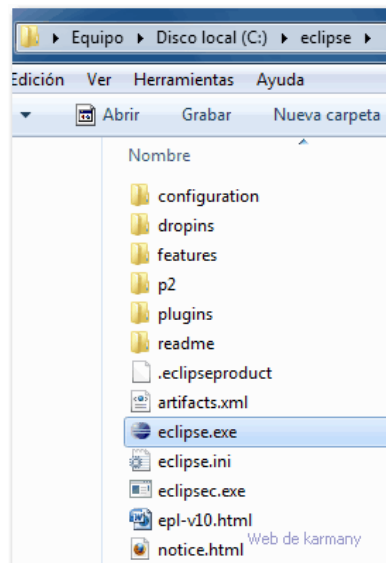
GRAFICO N° 3.7. Archivo zip de eclipse descomprimido



REALIZADO POR: La Autora

- 2) Una vez pegada la carpeta en el disco C:\ ejecutar el **eclipse.exe**:

GRAFICO N° 3.8. Ejecutar archivo eclipse.exe



REALIZADO POR: La Autora

- 3) Al ejecutar el archivo empezara a trabajar en el área de trabajo de eclipse.
- 4) Le pedirá que seleccione un área de trabajo [dejar la carpeta por Default], **OK** y listo para crear un nuevo proyecto android

GRAFICO N° 3.9. Pantalla bienvenida de eclipse



REALIZADO POR: La Autora

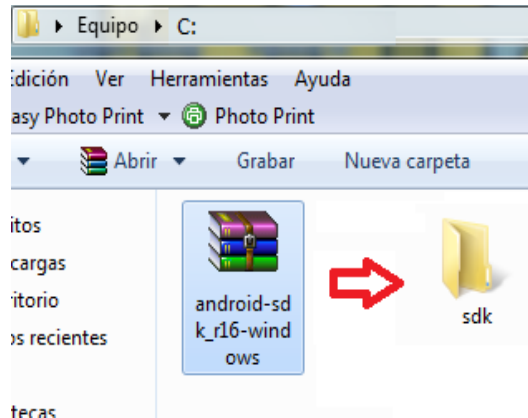
3.6.1.3. SDK

“Es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware. Es la maquina virtual de android para java”

INSTALACIÓN SDK MANAGER

- 1) Descomprimir el archivo descargado, pegar la carpeta **Sdk** en el disco **C:**

GRAFICO N° 3.10. Archivo zip de sdk descomprimido

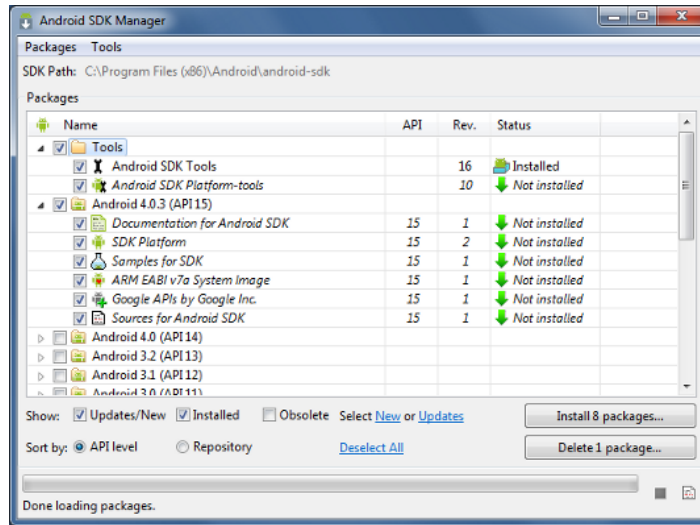


REALIZADO POR: La Autora

- 2) Abrir la carpeta y clic en **SDK de Manager.exe**
- 3) Clic en **Instalar** para los paquetes recomendados. El Administrador de Android SDK instala los paquetes seleccionados en su entorno de Android SDK.

Nota: Para la instalación de los paquetes la computadora debe tener conexión a internet para las descargas.

GRAFICO N° 3.11. Instalacion paquetes necesarios del **sdk manager.exe**

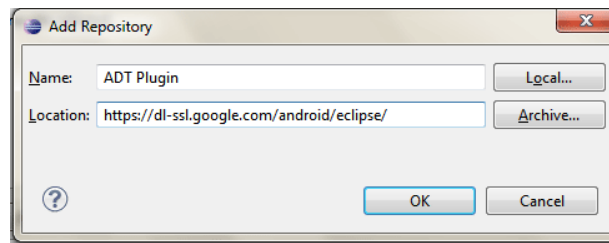


REALIZADO POR: La Autora

INSTALACIÓN PLUGIN ADT ANDROID PARA ECLIPSE

- 1) Abrir eclipse menu **Help->Install New Software**. Clic en **Add** y agregar en:
Name: Android ADT
Location: <https://dl-ssl.google.com/android/eclipse/> Y **OK**.

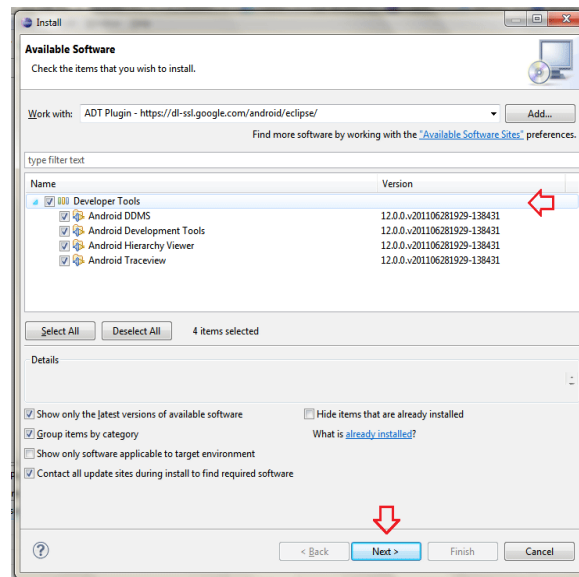
GRAFICO N° 3.12. Configuración plugin de android para eclipse



REALIZADO POR: La Autora

- 2) Aparecerá una lista con el elemento "**Developer Tools**". Y las herramientas que serán descargadas. Clic en el botón **Next**.

GRAFICO N° 3.13. Lista de herramientas a descargarse para el plugin



REALIZADO POR: La Autora

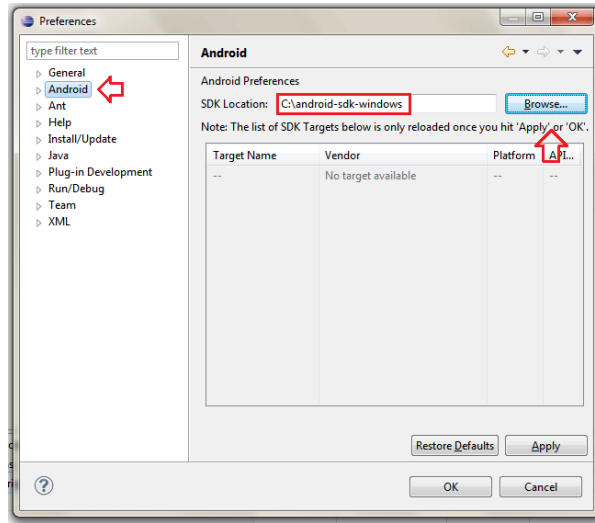
- 3) Aceptar los acuerdos de la licencia **Finish**, **OK** y reiniciar Eclipse.

CONFIGURACIÓN PLUGIN ADT EN ECLIPSE

Abrir eclipse para configurar el ADT y poder actualizar los datos de donde esta instalado el sdk.

- 1) Abrir eclipse menu **Window ->Preferences ->opción Android**
En **SDK Location:** Introducir la ruta donde se pego la carpeta SDK de Android.
- 2) Clic en **Apply** y luego **OK**.

GRAFICO N° 3.14. Actualizar el adt android en eclipse



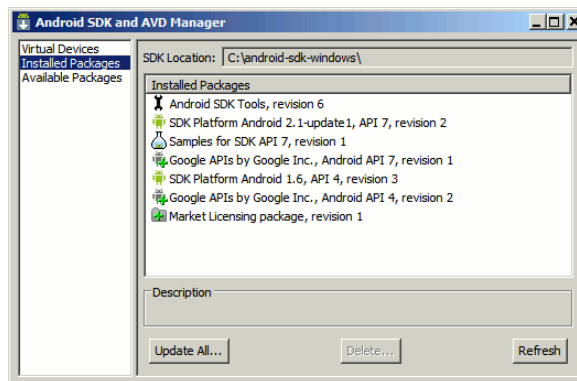
REALIZADO POR: La Autora

INSTALACIÓN DE TARGETS NECESARIOS.

Además del SDK de Android, también instalar SDK Targets de Android, que son las librerías necesarias para desarrollar en cada una de las versiones concretas de Android. Descargar su target correspondiente.

1. Abrir Eclipse menú “**Window / Android SDK and AVD Manager**”
2. Opcion Available Packages seleccionar e instalar todos los paquetes deseados.

GRAFICO N° 3.15. Instalacion targets de SDK



REALIZADO POR: La Autora

CONFIGURACION AVD MANAGER

Finalmente, crear un dispositivo virtual de Android que emule el funcionamiento del mismo. De esta forma se podrá ejecutar las aplicaciones.

- 1) Abrir Eclipse menú **Window->Android SDK and AVD Manager** o desde el siguiente icono en la barra de herramientas de Eclipse:

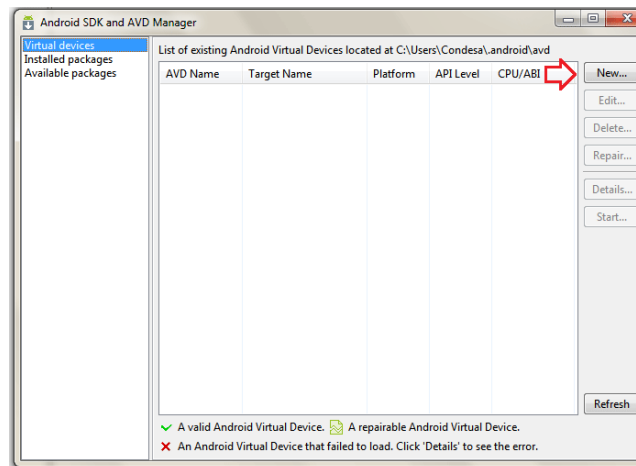
GRAFICO N° 3.16. Ejecutar e instalar el AVD manager.exe



FUENTE: La A utora

- 2) Seleccionar la opción **Virtual devices** y clic sobre el botón New

GRAFICO N° 3.17. Creando un nuevo dispositivo virtual AVD

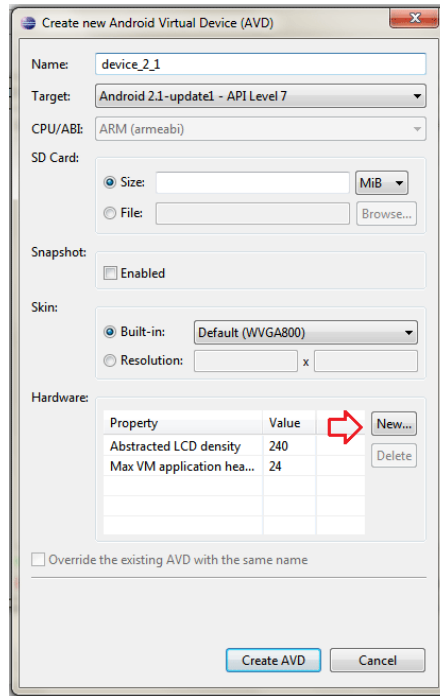


REALIZADO POR: La Autora

- 3) En la ventana **Create new Android Device (AVD)** llenar los datos:
Name: android Device
Target: Indica la versión de Android a emular 2.3.3.
SD Card: Establece la capacidad de la tarjeta de memoria indicar 32MB

4) Clic en **Créate Avd** para finalizar

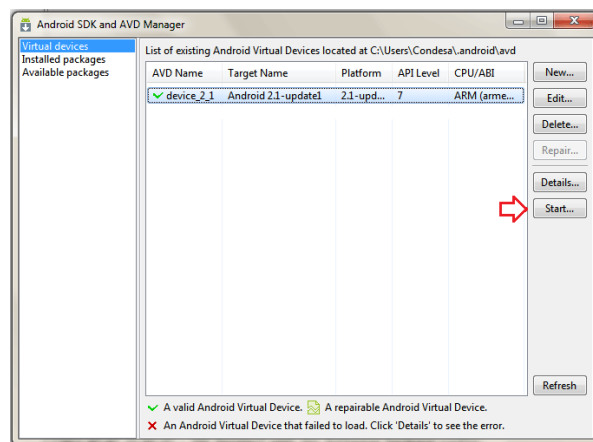
GRAFICO N° 3.18. Configurando el AVD



REALIZADO POR: La Autora

5) Aparecerá la siguiente pantalla. Para iniciar, dar clic sobre el botón **Start**

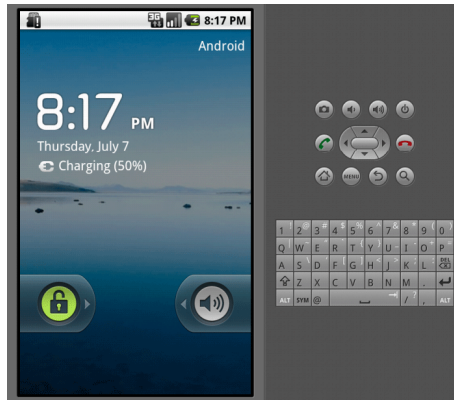
GRAFICO N° 3.19. Inicializando el AVD



REALIZADO POR: La Autora

- Una vez configurado todo aparecerá la pantalla del dispositivo virtual donde se ejecutaran las aplicaciones

GRAFICO N° 3.20. Pantalla inicial del emulador



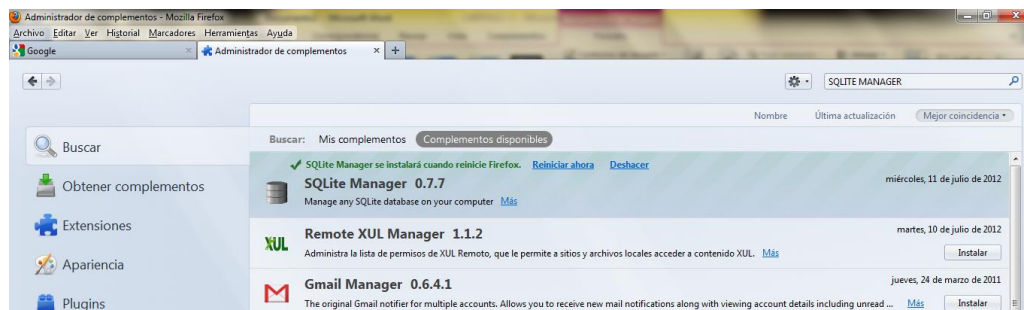
REALIZADO POR: La Autora

3.6.1.4. BASE DE DATOS SQLite

“SQLite es un motor de base de datos relacionales gratis y de código abierto. No necesita de un servidor para comunicarse con los datos sino que es directamente entre la aplicación y los datos, es portable.”

- 1) Abrir Firefox menu ->**herramientas** escribir el nombre del plugin sqlite una vez encontrada instalar y saldra una pantalla como la siguiente luego reiniciar el firefox

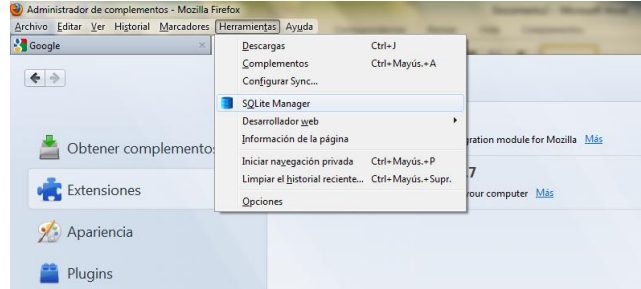
GRAFICO N° 3.21. Descargando el sqlite manager para Firefox



REALIZADO POR: La Autora

- 2) Para abrir la base menú **Herramientas**->SQLite manager

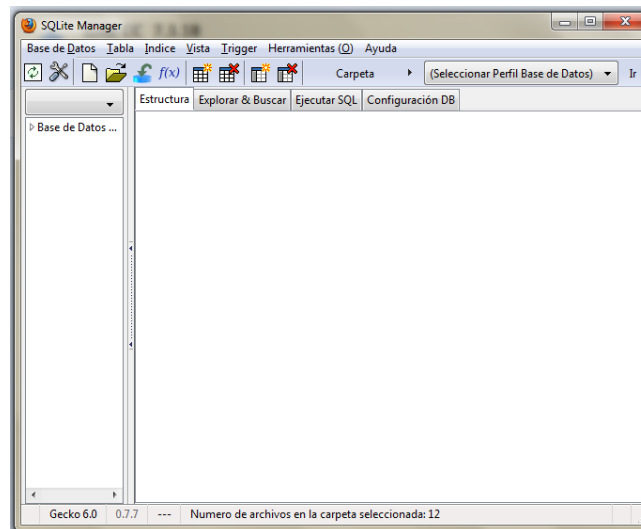
GRAFICO N° 3.22. Abrir base de datos sqlite



REALIZADO POR: La Autora

- 3) Y se tendrá el área de trabajo de sqlite.

GRAFICO N° 3.23. Área de trabajo de sqlite



REALIZADO POR: La Autora

3.6.1.5. DATAFRAMEWORK

“En cada actividad donde se quiera usar la base de datos se debe cargar la librería. Es importante que siempre se cargue la librería al empezar el método onCreate y cerrarla en el onDestroy”

En el onCreate sería así ("may.android.proyectogrado"):

```
try {
    DataFramework.getInstance().open(this, " may.android.proyectogrado ");
} catch (Exception e) {
    e.printStackTrace();
}
El onDestroy sería de la siguiente manera
@Override
protected void onDestroy() {
    super.onDestroy();
    DataFramework.getInstance().close();
}
```

3.6.2. Fase de Planificación

En la fase de planificación, los desarrolladores y clientes establecen los tiempos que tarda en realizar cada historia de usuario y la prioridad con la que serán implementadas. El cliente decide las historias que se seleccionarán para cada iteración. En la primera iteración se tratará de asentar la arquitectura del proyecto y en las subsecuentes iteraciones se irán incluyendo las historias de usuario seleccionadas. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración.

3.6.2.1. **Ámbito**

La situación de inicio es un escenario en el cual jóvenes estudiantes de la Carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi, desconocen el funcionamiento y el desarrollo de aplicaciones móviles con Android, al no autoeducarse ya sea por desinterés en conocer nuevas tecnologías en nuestro medio y especialmente la tecnología móvil. Es por esta razón que la presente aplicación servirá de apoyo en el aprendizaje de jóvenes estudiantes que deseen incursionar en el desarrollo de aplicaciones móviles bajo Android. Esta aplicación tomara el nombre de Aplicación de Entertentimiento (Mida su Inteligencia)

3.6.2.2. **Obtener requisitos**

3.6.2.2.1. **Identificación de los interesados**

En esta etapa se logró identificar a todos los interesados que directa o indirectamente se beneficiaran con el desarrollo de la aplicación.

A continuación se detalla una lista de las personas que contribuyeron en la obtención de requisitos:

- Docentes de la Carrera de CIYA
- Estudiantes de primero a octavos ciclos de la Carrera de CIYA

3.6.2.2.2. **Identificación de Actores**

Aquí se listan los usuarios potenciales del sistema, a continuación se muestra un cuadro identificando los actores de la aplicación:

TABLA N° 3.1. Actores de la aplicación

Requerimientos	Prioridad	Estado	Dificultad	Location	Subsistema
ACT 1: Docentes	Alta	Aprobado	Alta	D. Visión	Administrador
ACT 2: Estudiantes	Alta	Aprobado	Alta	D. Visión	Administrador

REALIZADO POR: La Autora

3.6.2.2.3. Requerimientos Informales

1. En la aplicación debe existir una pantalla inicial de presentación esta contendrá: autor de la aplicación, versión de software y una pequeña introducción.
2. Antes de iniciar el juego se debe mostrar un menú de opciones para la administración de los diferentes componentes de la aplicación como pueden ser: **Jugador, Preguntas, Jugar, Resumen, Acerca de, Instrucciones**, sin estar sujetos a estas, ya que pueden existir modificaciones o alteraciones durante el desarrollo de la aplicación.
3. En la opción **JUGADOR**, debe permitir gestionar los datos del mismo:
 - a. Ingresar datos del jugador, y luego regresar a un listado de jugadores para su verificación de ingreso.
 - b. Consultar jugadores, al momento de obtener un listado de jugadores, se debe implementar que al mantener presionado sobre el jugador nos despliegue dos opciones de eliminar y editar los datos del jugador.
 - c. Modificar datos de jugador, y luego regresar al listado de jugadores para verificar su modificación.
 - d. Eliminar datos de jugador, y luego regresar al listado de jugadores para su verificar su eliminación, considerando que al eliminar un jugador se deberán eliminar los datos que tengan relación a él.
4. La opción **PREGUNTAS**, permitirá gestionar datos de:
 - a. Ingreso de preguntas con su respuesta verdadera, para luego ser comparada con el test que genere o responda el jugador, éstas se desplegarán en una lista para su verificación de ingreso.
 - b. Al momento de obtener el listado de preguntas se debe implementar que al mantener presionado sobre la pregunta nos despliegue dos opciones de eliminar y editar la pregunta.
 - c. Modificar pregunta, y luego regresar al listado de preguntas para verificar su modificación

- d. Eliminar pregunta, y luego regresar al listado de preguntas para su verificar su eliminación
 - e. Ingreso de diferentes opciones de respuesta, implementando un botón que realice esta acción, se deberá escoger la pregunta y por cada ingreso de opción guardar, estas se desplegaran en una lista para su verificación de ingreso,
 - f. Al momento de obtener el listado de opciones de la pregunta seleccionada si se ingreso una opción mal implementar que al dar clic sobre la opción se la pueda eliminar.
5. La opción **JUGAR**, deberá permitir tener acceso al juego de mida su inteligencia a través de preguntas seleccionadas randómicamente de un banco de preguntas. Considerando:
- a. Al final del juego se emitirá un factor (%) de cumplimiento del test
 - b. Se deberá mostrar el resumen del juego, es decir cuantas preguntas malas, y cuantas preguntas buenas contesto el jugador, y estos datos se deberán almacenar para posterior análisis.
6. La opción **RESUMEN**, deberá acceder a un listado de jugadores, con los cuales al dar clic en uno de ellos se desplegará el resumen del juego, considerando que un jugador puede haber realizado varias partidas, y el resumen consistirá en visualizar cuantas preguntas malas y cuantas preguntas buenas generó en el test y su puntaje obtenido o valoración final.
7. La opción **ACERCA De**, se deberá visualizar información importante para el jugador que ofrezca un panorama claro del desarrollo de la aplicación como:
- a. Derechos de autor (información legal)
 - b. Fecha de desarrollo
 - c. Versión de la aplicación
 - d. Versión del Sistema Operativos sobre cual se debe ejecutar la aplicación
 - e. Versión de Base de Datos y del Kernel
8. En la opción **INSTRUCCIONES**, se dará a conocer al jugador las instrucciones de juego y lo que debe hacer antes de jugar

3.6.2.2.4. Historias de Usuarios o especificación de requisitos (Requerimientos)

Las historias de usuarios son tarjetas escritas por el cliente en un lenguaje natural, indicando las necesidades que el sistema debe satisfacer. A continuación se describen las historias de usuario finales.

TABLA N° 3.2. Primera historia de usuario(Introduccion)

Historia de Usuario	
Número: 01	Usuario: Administrador
Nombre de Historia: Introducción a la aplicación	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 30	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Se debe diseñar una pantalla de introducción a la aplicación esta contendrá autor de la aplicación, versión de software y una pequeña introducción.	
Observaciones: El programador de la aplicación es el que realiza esta acción.	

REALIZADO POR: La Autora

TABLA N° 3.3. Segunda historia de usuario (Gestion Menu de Opciones)

Historia de Usuario	
Número: 02	Usuario: Administrador
Nombre de Historia: Gestión Menú de Opciones (Programación)	
Prioridad en Negocio: Alta	Prioridad en Negocio: Alta
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Se mostrara un menú de opciones los cuales contendrán: Jugador, Preguntas, Jugar, Resumen, Acerca de, Instrucciones	
Observaciones: El programador de la aplicación es el que realiza esta acción.	

REALIZADO POR: La Autora

TABLA N° 3.4. Tercera historia de usuario(Gestion Jugadores)

Historia de Usuario	
Número: 03	Usuario: Administrador
Nombre de Historia: Gestión de Jugadores	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción JUGADOR el administrador del juego creara un nuevo jugador con su información para ser almacenado en la base de datos para su posterior uso, se consultara un listado de todos los jugadores existentes, así mismo podrá editar y eliminar la información de un jugador.	
Observaciones: El administrador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.5. Cuarta historia de usuario(Gestion Preguntas)

Historia de Usuario	
Número: 04	Usuario: Administrador
Nombre de Historia: Gestión Banco de Preguntas	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 40	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción PREGUNTAS el administrador puede ingresar determinadas preguntas con sus respectivas opciones de respuesta, estas serán verificadas o comparadas con la respuesta que dé el jugador a la misma, a la vez se mostrara una lista de todas las preguntas existentes en la base de datos.	
Observaciones: El administrador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.6. Quinta historia de usuario(Gestion Jugar)

Historia de Usuario	
Número: 05	Usuario: Jugador
Nombre de Historia: Gestión Jugar Aplicación	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 30	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción JUGAR el jugador podrá acceder al juego de mida su inteligencia, para esto deberá elegir su nombre de jugador, el botón continuar mostrara un paquete de preguntas seleccionadas randómicamente del banco de preguntas las cuales deberá responder.	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.7. Sexta historia de usuario(Gestion jugar II)

Historia de Usuario	
Número: 05.2	Usuario: Jugador
Nombre de Historia: Gestión Jugar Aplicación – Resultados Obtenidos	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Al final del juego se le mostrara los resultados obtenidos cuantas preguntas buenas y preguntas malas obtuvo	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.8. Séptima historia de usuario(Gestion Jugar III)

Historia de Usuario	
Número: 05.2	Usuario: Jugador
Nombre de Historia: Gestión Jugar Aplicación- Terminación del juego	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Al final de haber respondido las preguntas los resultados se almacenaran en una base de datos el cual contendrá: fecha en la que jugó, nombre del jugador y el resultado obtenido.	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.9. Octava historia de usuario(Gestion Resumen)

Historia de Usuario	
Número: 06	Usuario: Jugador
Nombre de Historia: Gestión Resumen de Juego	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 40	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción RESUMEN el jugador accederá a un listado de jugadores, deberá escoger su nombre de jugador y visualizar el resultado que obtuvo en su partida, ahí sabrá cuales fueron sus respuestas correctas e incorrectas.	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.10. Novena historia de usuario(Gestion Acerca de)

Historia de Usuario	
Número: 07	Usuario: Jugador
Nombre de Historia: Gestión Acerca de	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción ACERCA DE el jugador podrá obtener información muy importante como: nombre del autor, nombre y versión de sistema operativo,	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

TABLA N° 3.11. Decima historia de usuario(Gestion Instrucciones)

Historia de Usuario	
Número: 08	Usuario: Jugador
Nombre de Historia: Gestión Instrucciones	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Horas Estimadas: 20	Iteración Asignada: 1
Programador Responsable: Maritza Maigualca	
Descripción: Mediante la opción INSTRUCCIONES el jugador podrá guiarse y seguir los pasos para iniciar una partida	
Observaciones: El usuario jugador es el que realiza esta acción	

REALIZADO POR: La Autora

3.6.2.3. Análisis de Información

3.6.2.3.1. Priorización y estimación

En la siguiente tabla se representa la estimación del tiempo que se requiere para la implementación y la prioridad de las distintas historias de usuario recolectadas y corregidas, tomando en cuenta que un día laborable para el equipo de desarrollo XP es de 4 horas.

TABLA N° 3.12. Priorizacion de Historias de Usuario

Historias de Usuarios	Tiempo Estimado		
	Prioridad	Días Estimados	Horas Estimadas
001. Administración de jugadores	Alta	3,5	14
003. Banco de Preguntas	Alta	3	12
004. Jugar Aplicación	Alta	10	40
005. Resultados Obtenidos	Alta	4	16
006. Termino del juego	Alta	2	8
007. Resumen de Juego	Alta	10	40
008. Acerca de	Alta	2	8
009. Instrucciones	Alta	2	8

REALIZADO POR: La Autora

3.6.2.3.2. Distribución Funcional

Se identificaron los distintos módulos de aplicación, con el propósito de tomar en cuenta la distribución funcional de las historias de usuario recolectadas en la elaboración del plan de entrega, a más prioridad y el tiempo de implementación estimado.

En la siguiente tabla se muestra la funcionalidad detallada como gestión.

TABLA N° 3.13. Módulos de aplicación (Distribución Funcional)

Modulo	Historias de usuario
Gestión de jugadores	001. Administración de jugadores
	002. Datos de Jugador
	003. Editar o Eliminar Jugador
Gestión de preguntas	004. Ingreso de preguntas
	005. Editar o Eliminar pregunta
	006. Ingreso de opciones a la pregunta
	007. Eliminar opción
Gestión jugar aplicación	008. Eleccion de jugador
	009. Enlistar preguntas para empezar
Gestión resumen de partida	010. Eleccion de jugador
	011. Elección de partida
	012. Resumen de juego
Gestión acerca de	013. Datos de desarrollo de la aplicación
Gestión instrucciones	014. Instructivo de uso de la aplicación antes de jugar

REALIZADO POR: La Autora

3.6.3. Fase de Iteraciones

La fase de iteraciones hacia la entrega incluye varias iteraciones del sistema antes de la primera entrega. La programación que se determinó en la etapa de planeación es dividida en un número de iteraciones para la entrega.

La primera iteración crea la arquitectura de todo el sistema; esto es logrado seleccionando las historias que hacen cumplir la estructura para todo el sistema. El cliente decide las historias seleccionadas para cada iteración. Al final de la última iteración, el sistema estará listo para ser entregado y llevarlo a producción.

3.6.3.1. Diseño

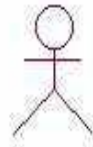
3.6.3.1.1. Diagramas de casos de uso

Los diagrama de casos de uso representan la forma en como un Cliente (Actor) opera con el sistema (aplicacion) en desarrollo, además de la forma, tipo y orden en como los elementos interactuan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

Actor:

GRAFICO N° 3.24. Elemento Actor de Caso de Uso

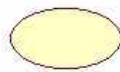


REALIZADO POR: La Autora

Un actor es un rol que un usuario juega con respecto al sistema. Un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Caso de Uso:

GRAFICO N° 3.25. Elemento caso de uso

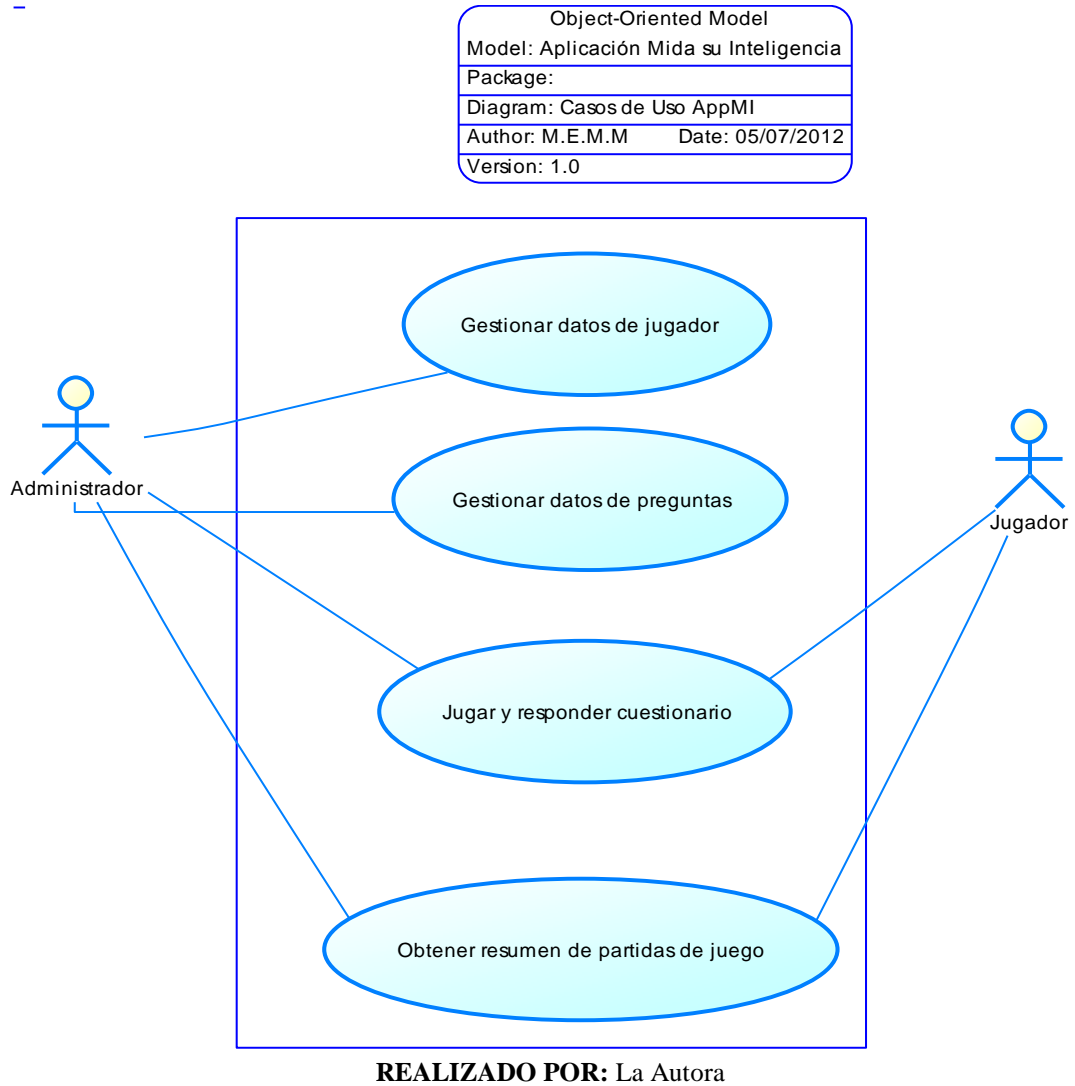


REALIZADO POR: La Autora

Es una operación/tarea específica que se realiza tras una orden o petición de un actor.

APLICACIÓN MOVIL MIDA SU INTELIGENCIA

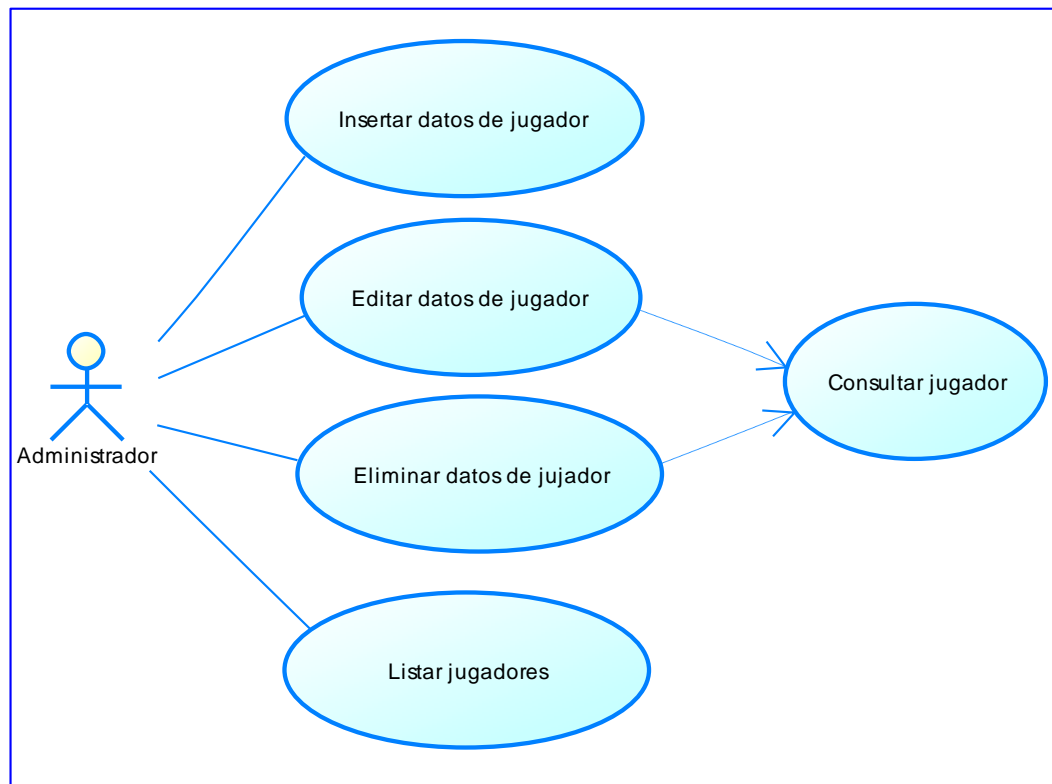
DIAGRAMA N° 3.1. Diagrama de casos de uso aplicación mida su inteligencia



GESTION DATOS DE JUGADOR

DIAGRAMA N° 3.2. Diagrama de casos de uso gestión jugador

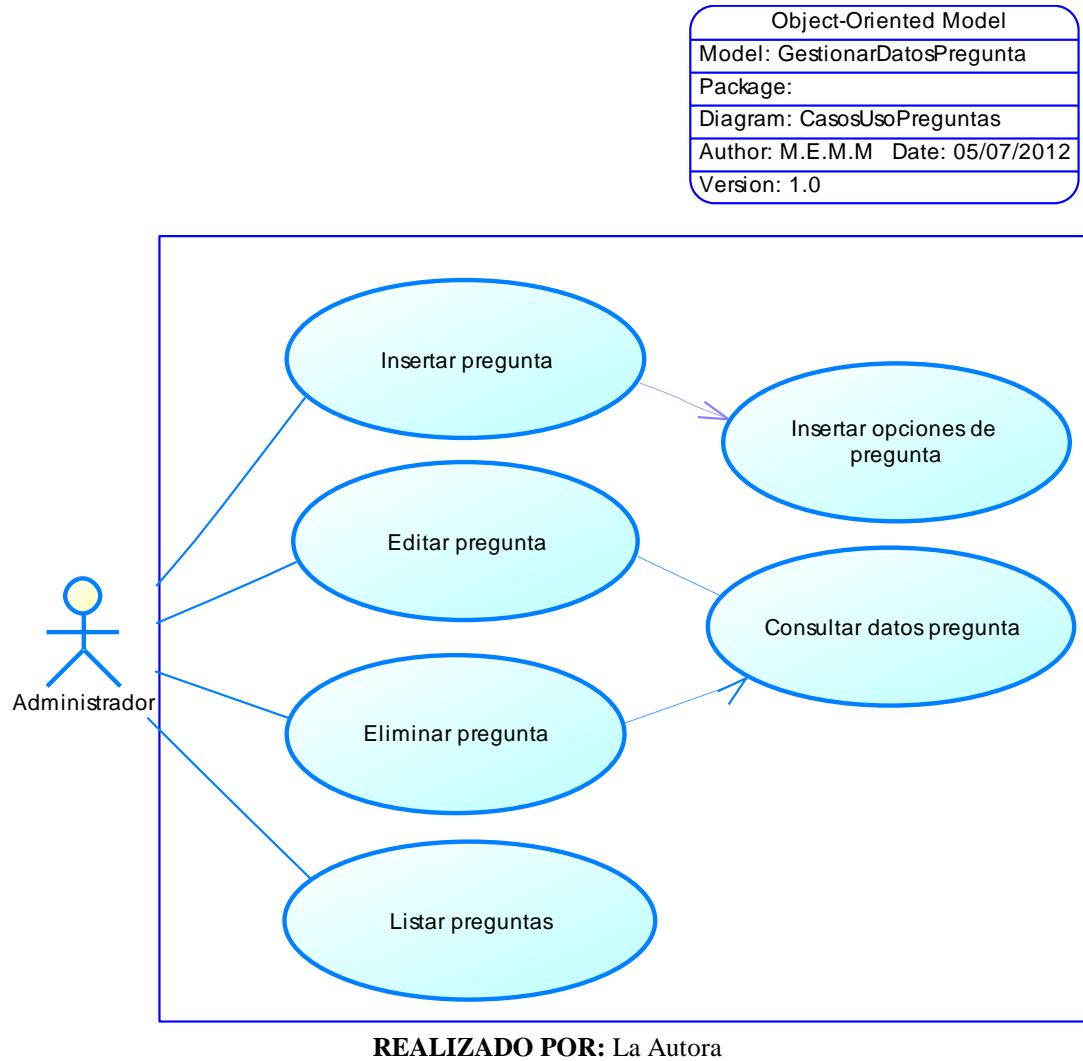
Object-Oriented Model
Model: Gestionar Datos de Jugador
Package:
Diagram: CasosUsoGestionJugador
Author: M.E.M.M Date: 05/07/2012
Version: 1.0



REALIZADO POR: La Autora

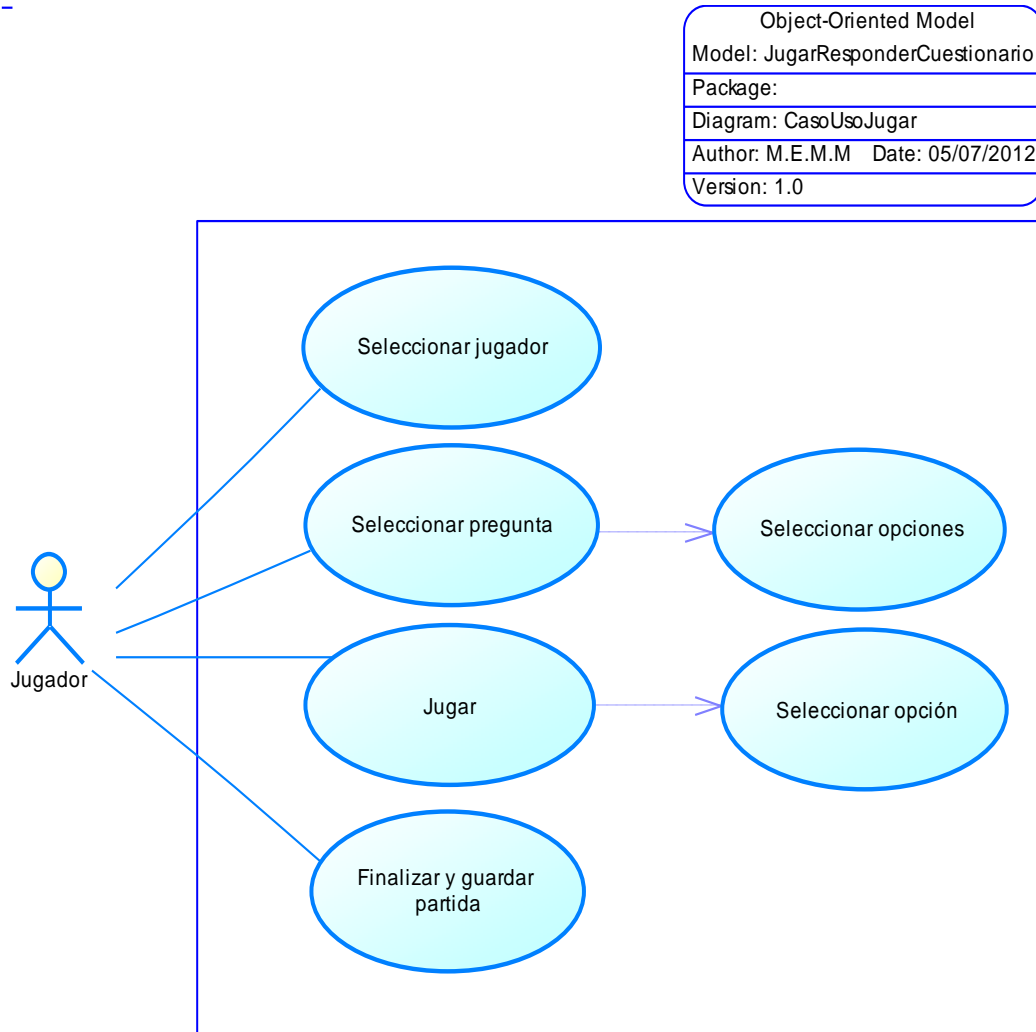
GESTION DATOS DE PREGUNTA

DIAGRAMA N° 3.3. Diagrama de casos de uso gestión preguntas



GESTION JUGAR Y RESPONDER CUESTIONARIO

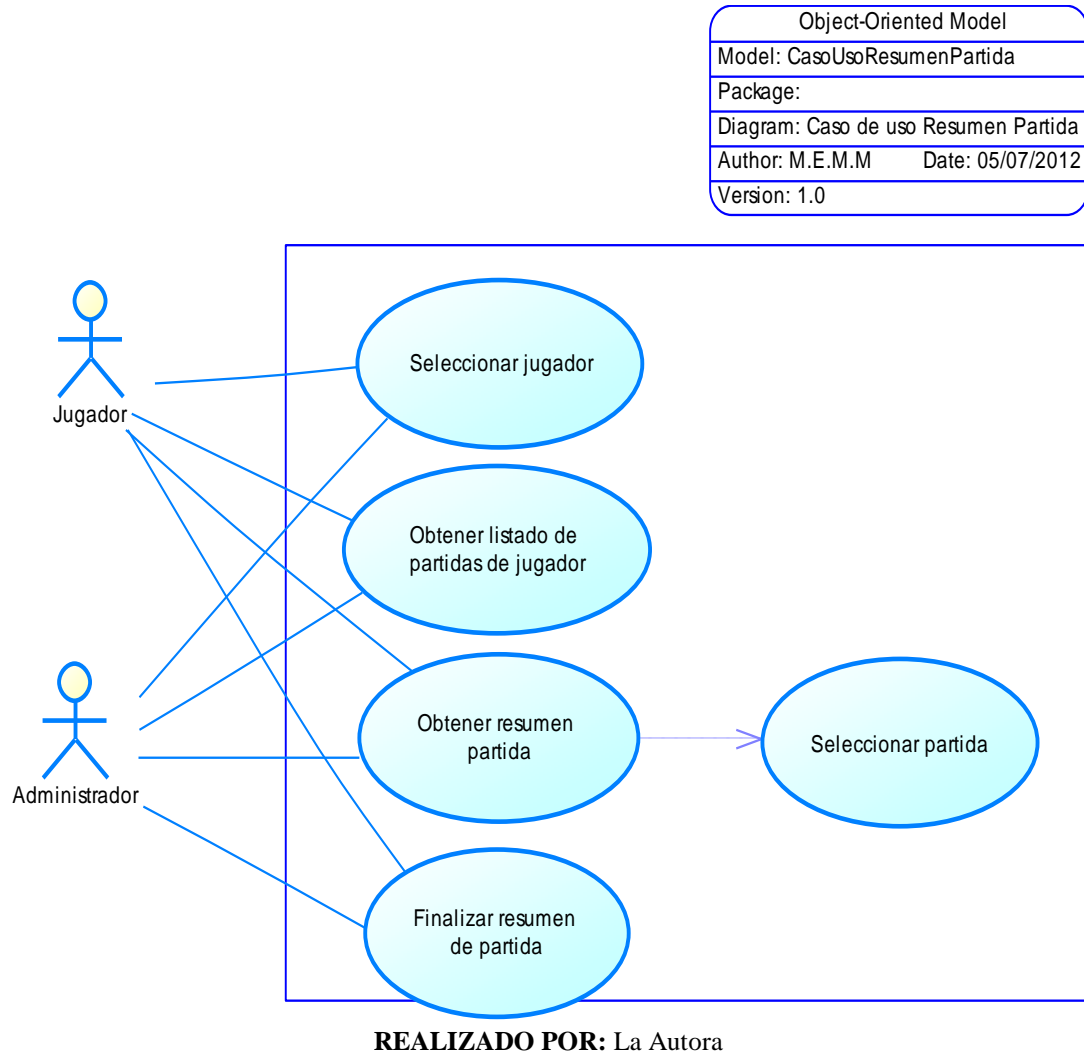
DIAGRAMA N° 3.4. Diagrama de casos de uso gestión jugar



REALIZADO POR: La Autora

GESTION OBTENER RESUMEN PARTIDA

DIAGRAMA N° 3.5. Diagrama de casos de uso gestión resumen partida



3.6.3.1.2. Diagrama de actividad

Los diagramas de actividad representan el comportamiento interno de una operación o de un caso de uso, bajo la forma de un desarrollo por etapas, agrupadas secuencialmente.

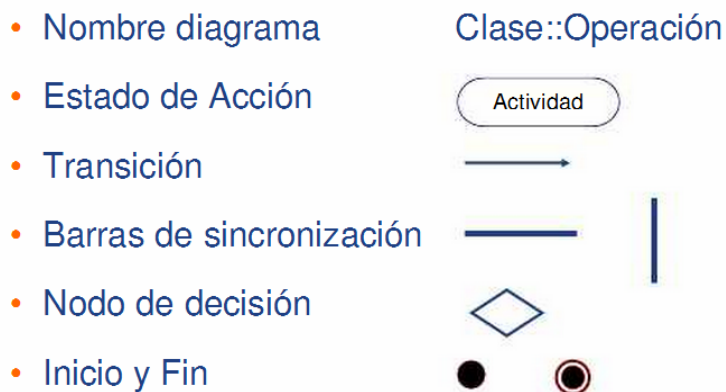
El propósito de los diagramas de actividad son:

Modelar el flujo de tareas

Modelar las operaciones

Los elementos principales de un diagrama de actividad se representan en el siguiente gráfico:

GRAFICO N° 3.26. Elementos de un Diagrama de Actividad



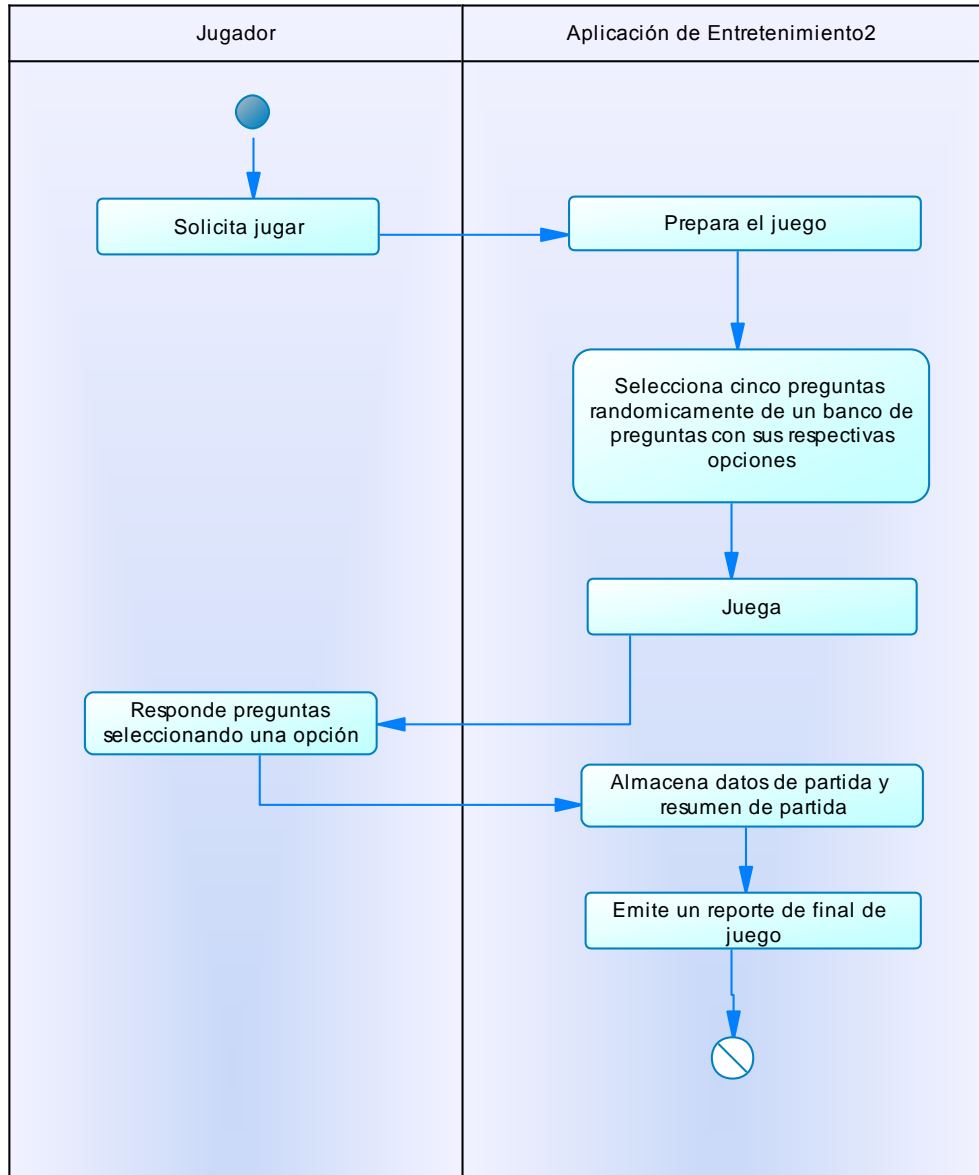
Fuente: <http://es.scribd.com/doc/2568098/UML-Diagramas-de-actividad>

Los diagramas de actividad obtenidos para la realización de la aplicación de entretenimiento “Midas Inteligencia” se muestran a continuación tanto para jugador como para el administrador del juego:

DIAGRAMA N° 3.6. Diagrama de actividad 1 (JUGADOR)

JUGADOR

Object-Oriented Model	
Model: Diagrama de Actividad Mida Su Inteligencia	
Package:	
Diagram: DiagramaActividadMidaInteligencia	
Author: M.E.M.M	Date: 10/07/2012
Version: 1.0	

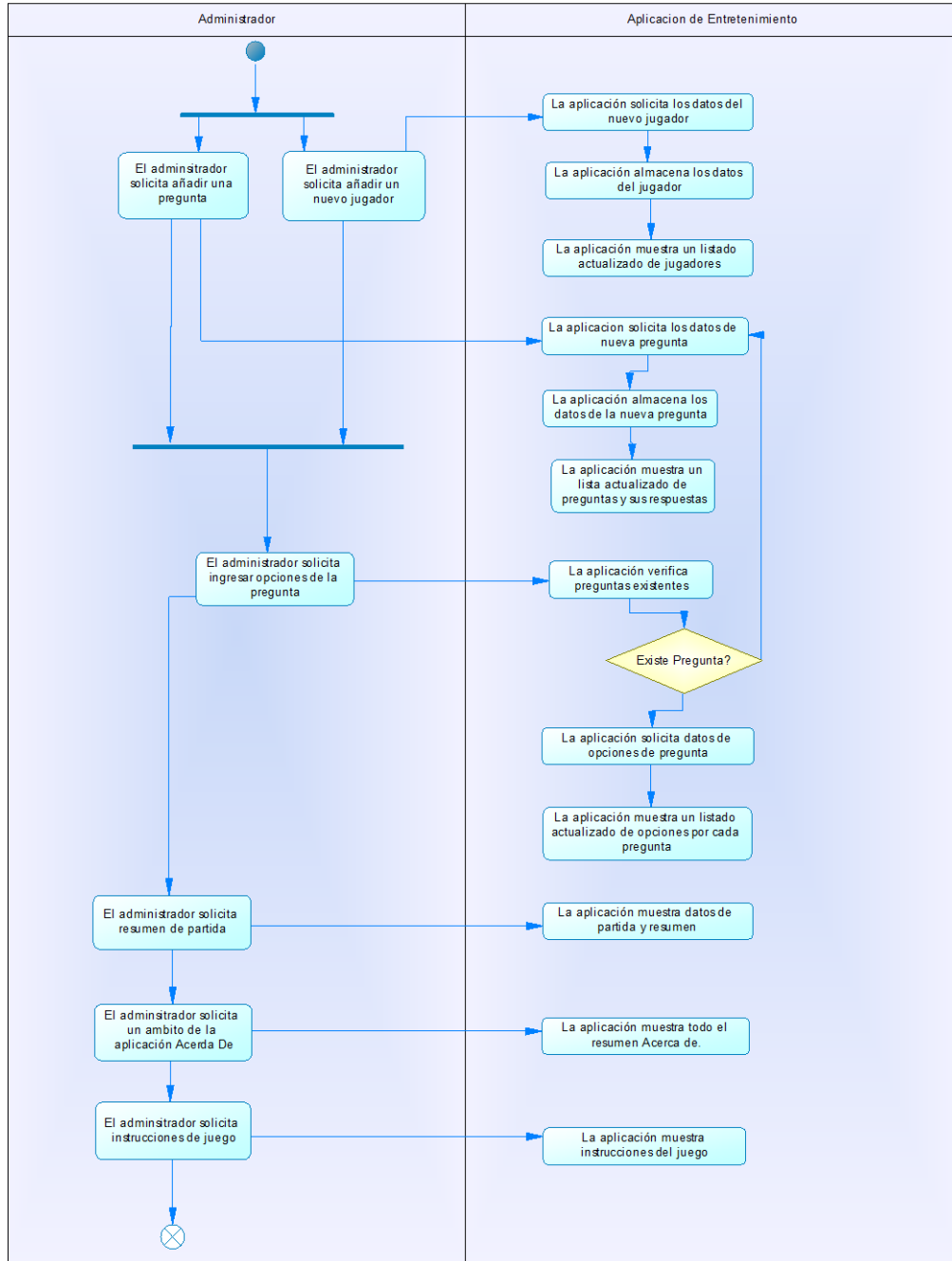


REALIZADO POR: La Autora

DIAGRAMA N° 3.7. Diagrama de actividad 2 (ADMINISTRADOR)

ADMINISTRADOR

Object-Oriented Model	
Model: Diagrama de Actividad Mida Su Inteligencia	
Package:	
Diagram: DiagramaActividadMidaInteligencia	
Author: M.E.M.M	Date: 10/07/2012
Version: 1.0	



REALIZADO POR: La Autora

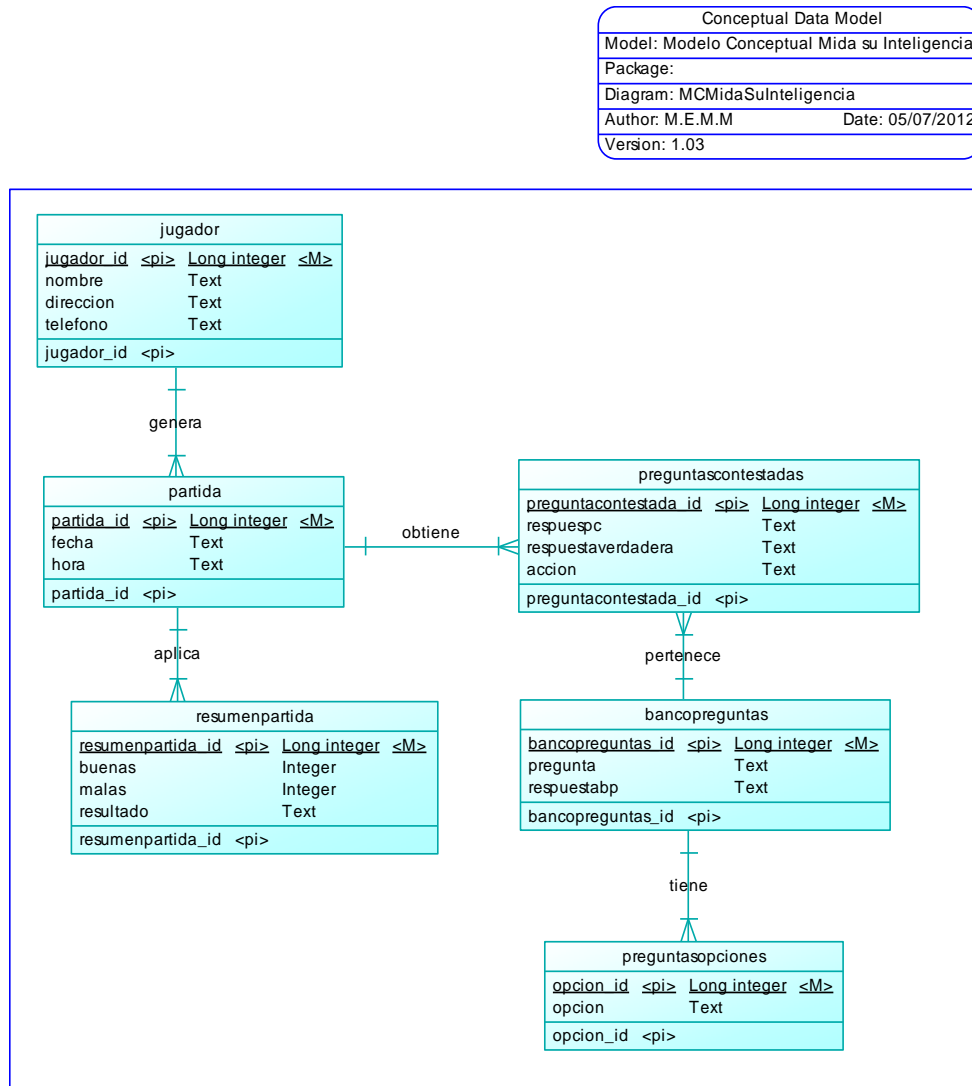
3.6.3.1.3. Diseño de datos

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos:

- La velocidad de acceso,
- El tamaño y tipo de información,
- Facilidad de acceso y extracción de a la información requerida,

MODELO CONCEPTUAL DE LA BASE DE DATOS

DIAGRAMA N° 3.8. Diagrama modelo conceptual de la base de datos

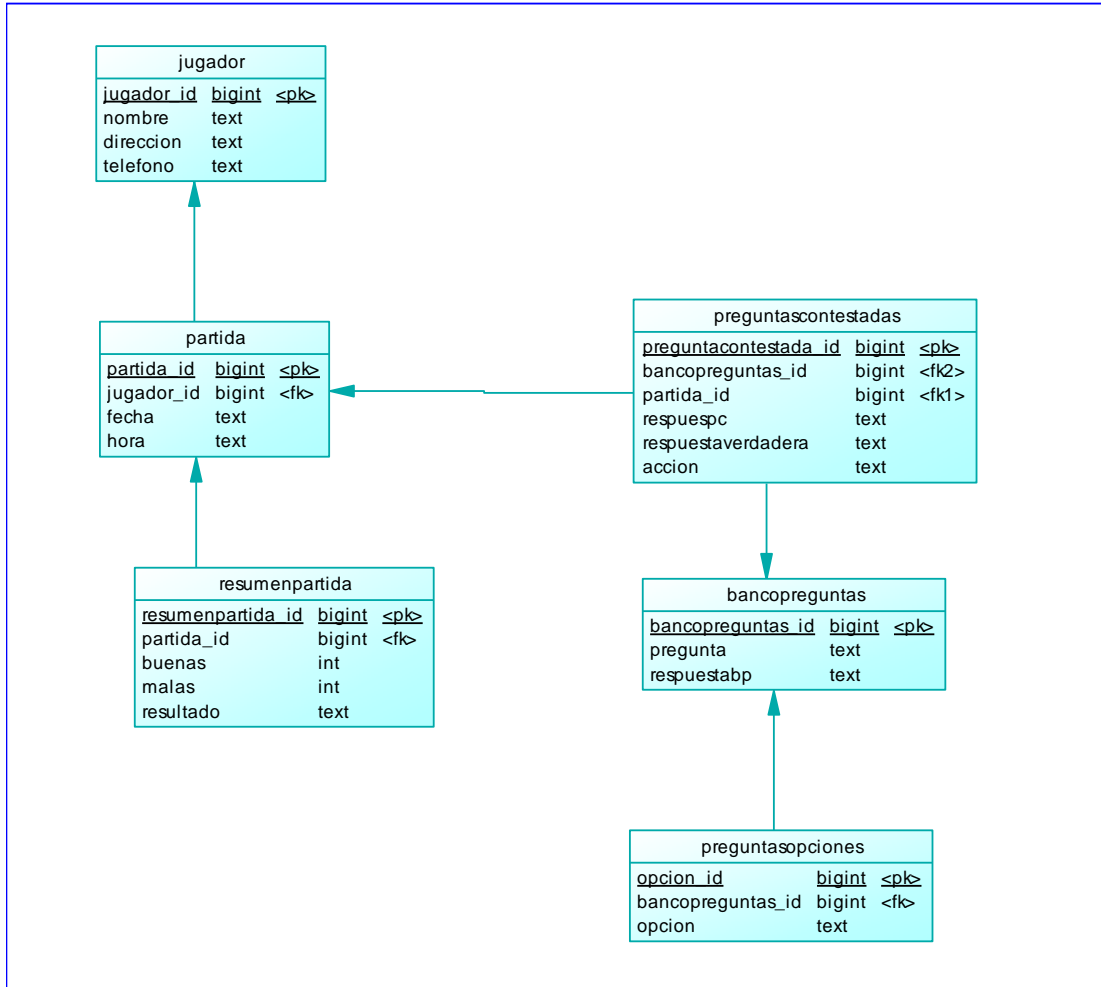


REALIZADO POR: La Autora

MODELO FÍSICO DE LA BASE DE DATOS

DIAGRAMA N° 3.9. Diagrama modelo físico de la base de datos

Physical Data Model	
Model: Modelo Físico Mida su Inteligencia	
Package:	
Diagram: MCMidaSuInteligencia	
Author: M.E.M.M	Date: 05/07/2012
Version: 1.03	



REALIZADO POR: La Autora

SCRIB BASE DE DATOS

```
/*=====*/
/* DBMS name:   SQLite 5.0                */
/* Created on:  05/07/2012 18:40:27       */
/*=====*/

drop table if exists BANCOPREGUNTAS;
drop table if exists JUGADOR;
drop table if exists PARTIDA;
drop table if exists PREGUNTASCONTESTADAS;
drop table if exists PREGUNTASOPCIONES;
drop table if exists RESUMENPARTIDA;
/*=====*/
/* Table: BANCOPREGUNTAS                    */
/*=====*/
create table BANCOPREGUNTAS
(
  BANCOPREGUNTAS_ID  bigint not null,
  PREGUNTA           text,
  RESPUESTABP       text,
  primary key (BANCOPREGUNTAS_ID)
);
/*=====*/
/* Table: JUGADOR                            */
/*=====*/
create table JUGADOR
(
  JUGADOR_ID        bigint not null,
  NOMBRE            text,
  DIRECCION         text,
  TELEFONO          text,
  primary key (JUGADOR_ID)
);
/*=====*/
/* Table: PARTIDA                            */
/*=====*/
create table PARTIDA
(
  PARTIDA_ID        bigint not null,
  JUGADOR_ID        bigint not null,
  FECHA             text,
  HORA              text,
  primary key (PARTIDA_ID)
);
```

```

/*=====*/
/* Table: PREGUNTASCONTESTADAS */
/*=====*/
create table PREGUNTASCONTESTADAS
(
  PREGUNTACONTESTADA_ID bigint not null,
  BANCOPREGUNTAS_ID  bigint not null,
  PARTIDA_ID        bigint not null,
  RESPUESPC         text,
  RESPUESTAVERDADERA text,
  ACCION            text,
  primary key (PREGUNTACONTESTADA_ID)
);
/*=====*/
/* Table: PREGUNTASOPCIONES */
/*=====*/
create table PREGUNTASOPCIONES
(
  OPCION_ID        bigint not null,
  BANCOPREGUNTAS_ID  bigint not null,
  OPCION           text,
  primary key (OPCION_ID)
);
/*=====*/
/* Table: RESUMENPARTIDA */
/*=====*/
create table RESUMENPARTIDA
(
  RESUMENPARTIDA_ID  bigint not null,
  PARTIDA_ID         bigint not null,
  BUENAS             int,
  MALAS              int,
  RESULTADO          text,
  primary key (RESUMENPARTIDA_ID)
);

alter table PARTIDA add constraint FK_GENERA foreign key (JUGADOR_ID)
  references JUGADOR (JUGADOR_ID) on delete restrict on update restrict;

alter table PREGUNTASCONTESTADAS add constraint FK_OBTIENE foreign key
(PARTIDA_ID)
  references PARTIDA (PARTIDA_ID) on delete restrict on update restrict;

alter table PREGUNTASCONTESTADAS add constraint FK_PERTENECE foreign key
(BANCOPREGUNTAS_ID)
  references BANCOPREGUNTAS (BANCOPREGUNTAS_ID) on delete restrict on
update restrict;

```

```
alter table PREGUNTASOPCIONES add constraint FK_TIENE foreign key
(BANCOPREGUNTAS_ID)
  references BANCOPREGUNTAS (BANCOPREGUNTAS_ID) on delete restrict on
update restrict;
```

```
alter table RESUMENPARTIDA add constraint FK_APLICA foreign key (PARTIDA_ID)
  references PARTIDA (PARTIDA_ID) on delete restrict on update restrict;
```

3.6.3.1.4. Mapa de la aplicación

- Ingreso a la aplicación

Introducción

Menú principal

Aplicación de Entretenimiento

Jugador

Preguntas

Jugar

Resumen

Acerca de

Instrucciones

3.6.3.2. Construcción

3.6.3.2.1. Diseño de la experiencia de usuarios

El objetivo de esta sección es que el cliente-usuario que use la aplicación se encuentre familiarizado con el procedimiento que desea realizar de una manera más simple y sencilla, de tal forma que se sienta cómodo y seguro de usarlo.

A continuación se presentan los prototipos de la aplicación:

3.6.3.2.2. Prototipo de interfaces

Esta pantalla es una breve introducción a la aplicación de Mida su Inteligencia. Al dar clic en el botón inicio aparecerá la pantalla de Menú Principal.

GRAFICO N° 3.27. Pantalla Inicial



REALIZADO POR: La Autora

Esta pantalla muestra un menu cada icono del menú realiza un trabajo específico a continuación se explicara cada una de sus funciones:

GRAFICO N° 3.28. Pantalla Menú Principal

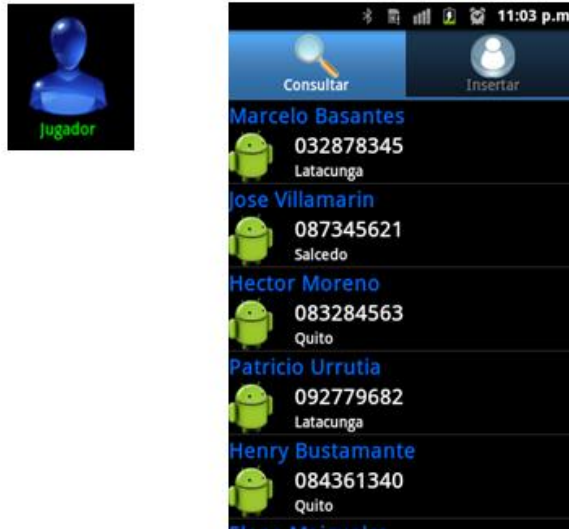


REALIZADO POR: La Autora

El primer icono pertenece a **Jugador** este tendra dos opciones:

Una de consultar jugador saldra una lista detallada de los jugadores existentes, ai no existe debe crearlo en la opcion de **Insertar**.

GRAFICO N° 3.29. Pantalla Jugador (consultar jugador)



REALIZADO POR: La Autora

La segunda opción **Insertar** pedirá el nombre, dirección y teléfono del jugador y clic en grabar.

GRAFICO N° 3.30. Pantalla Jugador (insertar jugador)

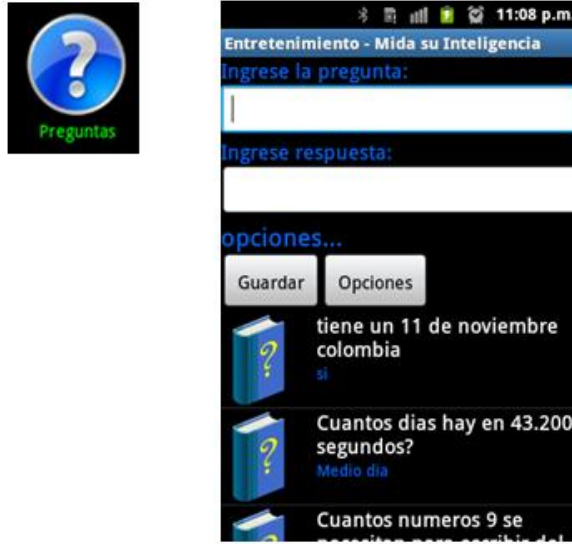


REALIZADO POR: La Autora

El segundo icono pertenece a **Preguntas** aquí se debe llenar un banco de preguntas con sus respectivas opciones.

Se ingresa la pregunta y la respuesta correcta y clic en guardar, ira apareciendo en la lista todas las preguntas que se vaya ingresando.

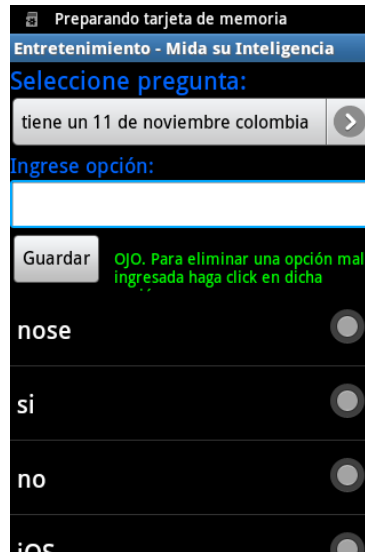
GRAFICO N° 3.31. Pantalla Preguntas (ingreso de preguntas y respuesta)



REALIZADO POR: La Autora

Para ingresar las opciones a esa pregunta clic en el botón opciones ira a una pantalla donde se debe escoger la pregunta y llenar las opciones de respuesta una por una y dar clic en guardar.

GRAFICO N° 3.32. Pantalla Preguntas (Ingreso de opciones a cada pregunta)



REALIZADO POR: La Autora

El tercer icono pertenece a **Jugar** aquí empezara el juego. Le saldrá una pantalla pidiendo si ha leído las instrucciones de uso de juego si elige si deberá escoger su nombre de jugador y clic en el botón continuar para empezar

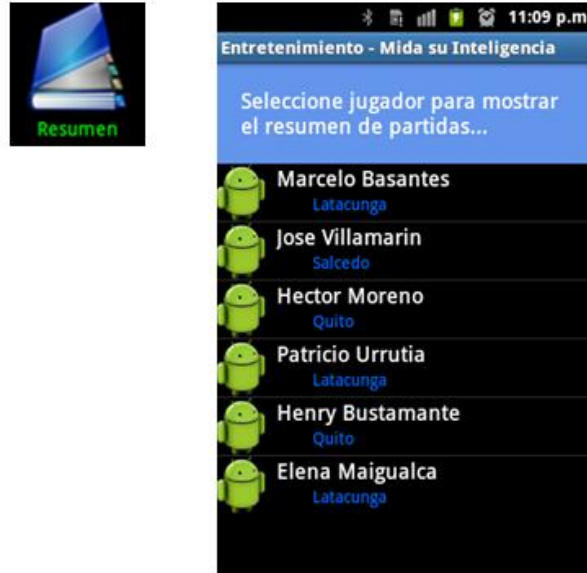
GRAFICO N° 3.33. Pantalla jugar



REALIZADO POR: La Autora

El cuarto icono pertenece a **Resumen de Partida** aquí saldrá una lista detallada de todos los jugadores existentes clic sobre el nombre de jugador

GRAFICO N° 3.34. Pantalla Resumen (resumen lista jugadores)



REALIZADO POR: La Autora

Saldrá una pantalla indicando los datos del jugador y una lista detallada de todas las partidas que tenga el jugador. Dar clic sobre la partida que se desea saber los resultados obtenidos

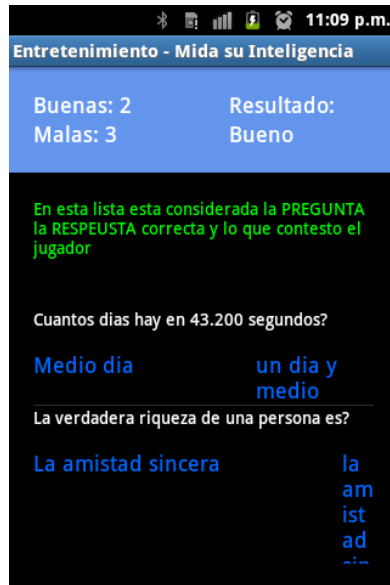
GRAFICO N° 3.35. Pantalla Resumen (resumen lista partida)



REALIZADO POR: La Autora

De inmediato saldrá una pantalla indicando los resultados obtenidos de la partida

GRAFICO N° 3.36. Pantalla Resumen (resumen de juego)



REALIZADO POR: La Autora

El quinto icono pertenece a **ACERCA DE** aquí se indica pequeños datos técnicos de autoría de la aplicacion

GRAFICO N° 3.37. Pantalla Acerca de (datos de autoria de aplicacion)



REALIZADO POR: La Autora

El sexto icono pertenece a **Instrucciones** de uso de la aplicación y pasos a seguir para iniciar una partida

GRAFICO N° 3.38. Pantalla Instrucciones (instrcciones a seguir para iniciar juego)



REALIZADO POR: La Autora

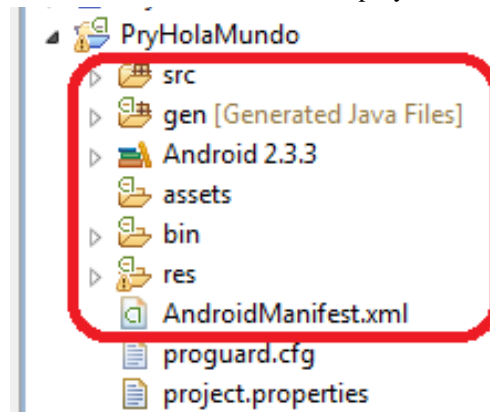
3.6.3.2.3. Codificación

Antes de iniciar con el proceso de desarrollo de aplicaciones móviles en este caso la aplicación de entretenimiento “Mida su Inteligencia”.

Se dejara claro algunas pautas que serán importantes en el proceso de desarrollo.

1. Una vez ya creado el proyecto Android
2. Es factible explicar la ESTRUCTURA DE UN PROYECTO ANDROID
3. Mi primera aplicación se llamara **hola mundo**

GRAFICO N° 3.39. Estructura proyecto android

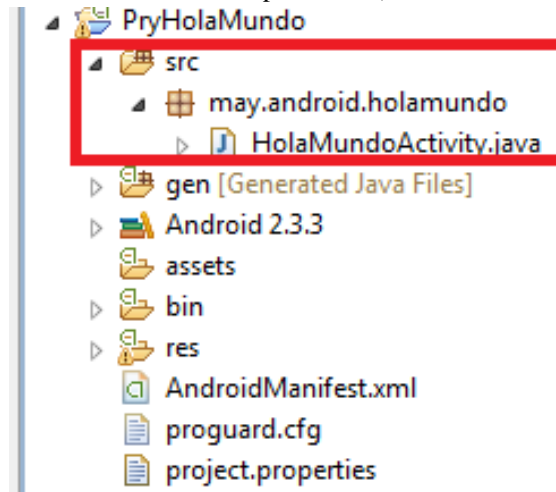


REALIZADO POR: La Autora

LA CARPETA SRC:

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

GRAFICO N° 3.40. La carpeta **SRC** (HolamundoActivity.java)



REALIZADO POR: La Autora

Código de la actividad (HolaMundoActivity.java)

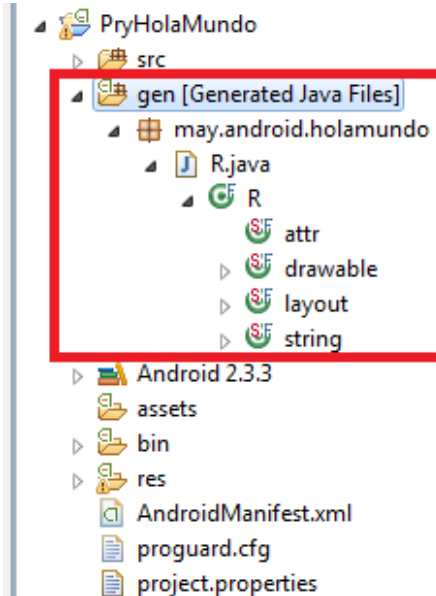
```
package may.android.holamundo;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
public class HolaMundoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

LA CARPETA GEN:

Contiene una serie de elementos de código java generados automáticamente al compilar un proyecto Android. El fichero R.java, y la clase R contendrán en todo momento una serie de constantes con los ID de todos los recursos de la aplicación, de forma que se pueda acceder fácilmente a estos recursos.

GRAFICO N° 3.41. La carpeta GEN



REALIZADO POR: La Autora

El código generado es el siguiente (R.JAVA)

```
package may.android.holamundo;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

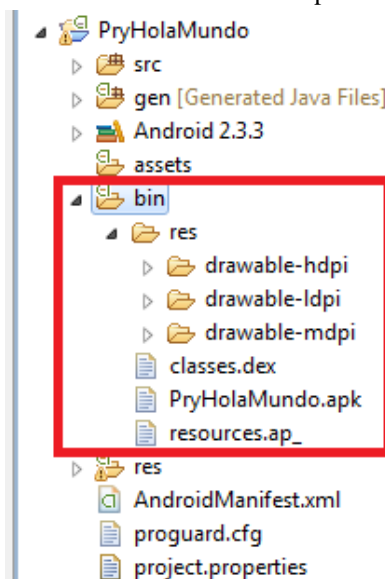
CARPETA /ASSETS/

Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc. Estos ficheros no generan un ID.

LA CARPETA BIN:

Contiene los archivos .dex código de la maquina virtual de android, un .apk de nuestro proyecto, y otros recursos de imágenes mapeadas

GRAFICO N° 3.42. La carpeta **BIN**



REALIZADO POR: La Autora

LA CARPETA RES:

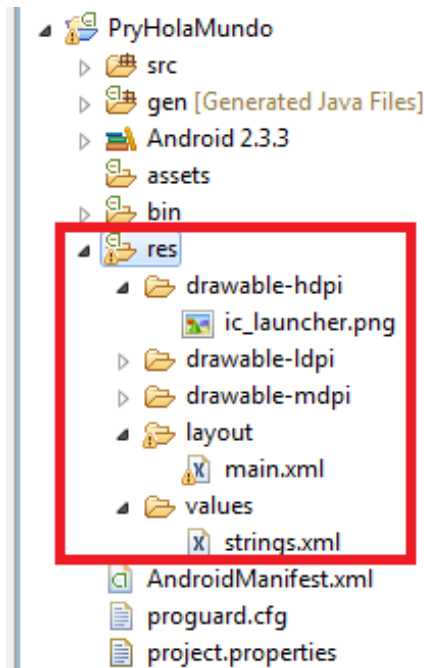
Contiene todos los recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos se distribuyen en las siguientes carpetas:

- **/res/drawable/**. Contiene las imágenes de la aplicación. Se divide en **/drawable-ldpi**, **/drawable-mdpi** y **/drawable-hdpi** para utilizar diferentes recursos dependiendo de la resolución del dispositivo.

- **/res/layout/**. Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Se puede dividir en /layout y /layout-land para definir distintos layouts dependiendo de la orientación del dispositivo.
- **/res/menu/**. Contiene la definición de los menús y menus contextuales de la aplicación.
- **/res/values/**. Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc.
- **/res/xml/**. Contiene los ficheros XML utilizados por la aplicación.
- **/res/raw/**. Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.

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

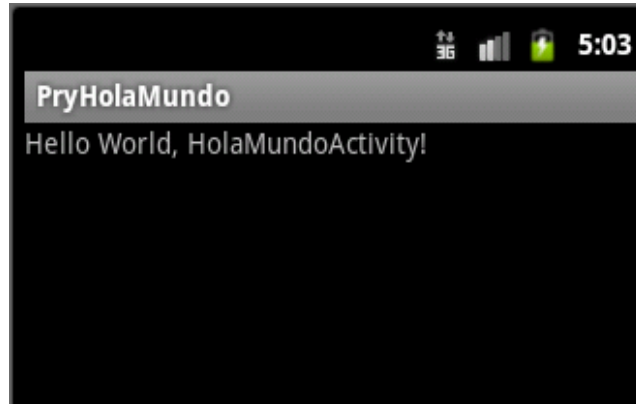
GRAFICO N° 3.43. La carpeta **RES**



REALIZADO POR: La Autora

/RES/LAYOUT/. Contiene un archivo de layout o entorno grafico main.xml (Vista)

GRAFICO N° 3.44. Interfaz grafica de proyecto



REALIZADO POR: La Autora

El codigo generado es el siguiente (main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="HOLA MUNDO ESTA ES MI PRIMERA ACTIVIDAD" />
</LinearLayout>
```

/RES/VALUES/. Contiene un archivo string de contenidos **string.XML**

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <string name="hello">Hello World, HolaMundoActivity!</string>
    <string name="app_name">PryHolaMundo</string>
</resources>
```

FICHERO ANDROIDMANIFEST.XML

Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono), sus componentes (pantallas, mensajes), o los permisos necesarios para su ejecución.

El código generado es (AndroidManifest.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="may.android.holamundo"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="10" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".HoLaMundoActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

NOTA:

Cabe recalcar que mientras se vaya creando una layout de diseño siempre se debe crear una clase de la misma ya que en la clase se genera el código más no en la layout.

No importa cuantas layout se tengan siempre anclarlas al archivo manifest ya que al momento de ejecutarlas no se podrá visualizar

APLICACIÓN DE ENTRETENIMIENTO

A continuación se presenta el código generado para la layout inicial a la aplicación

GRAFICO N° 3.45. Layout (main.xml)



REALIZADO POR: La Autora

ProyectoGradoActivity.java

```
package may.android.proyectogrado;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class ProyectoGradoActivity extends Activity {
    /** Called when the activity is first created. */
    Button btinicio;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btinicio=(Button) findViewById(R.id.btinicio);
        btinicio.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Intent intento=new Intent(ProyectoGradoActivity.this,Menu_Principal.class);
                startActivity(intento);
            }
        });
    }
}
```

An la layout anterior al dar clic sobre el botón inicio aparecerá la layout de menú principal, cada icono realia una función:

GRAFICO N° 3.46. Layout (menú_principal.xml)



REALIZADO POR: La Autora

Menu_Principal.java

```

package may.android.proyectogrado;
import android.app.Activity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
public class Menu_Principal extends Activity{
    GridView grid_main;
    String[] datos=new String[]{"Jugador","Preguntas","Jugar","Resumen","Acerca de","Instrucciones"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu_principal);

        grid_main = (GridView)findViewById(R.id.GridView01);
        grid_main.setAdapter((ListAdapter) new ImageAdapter(this,datos));
        grid_main.setOnItemClickListener(new OnItemClickListener(){
            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int position,long arg3) {
                // TODO Auto-generated method stub
                Toast.makeText(getApplicationContext()," " + position, Toast.LENGTH_SHORT).show();
                switch(position){
                    case 0:
                        Intent intento=new Intent(Menu_Principal.this,ContenidoTabs.class);
                        startActivity(intento);
                        break;
                    case 1:
                        Intent intento1=new Intent(Menu_Principal.this,Preguntas.class);
                        startActivity(intento1);
                        break;
                    case 2:
                        Intent intento2=new Intent(Menu_Principal.this,JugadorLayuot2.class);

```

```

        startActivity(intento2);
        break;
    case 3:
        Intent intento3=new Intent(Menu_Principal.this,ResumenListadoJugadores.class);
        startActivity(intento3);
        break;
    case 4:
        Intent intento4=new Intent(Menu_Principal.this,AcercaDe.class);
        startActivity(intento4);
        break;
    case 5:
        Intent intento5=new Intent(Menu_Principal.this,Instrucciones.class);
        startActivity(intento5);
        break;
    }
    });
}
//ADAPTADOR
public class ImageAdapter extends BaseAdapter{
    Context mContext;
    String[] datos1;
    // references to our images
    private Integer[] iconos = {
        R.drawable.sample_0, R.drawable.sample_1,
        R.drawable.sample_2, R.drawable.sample_3,
        R.drawable.sample_4, R.drawable.sample_5
    };
    public static final int ACTIVITY_CREATE = 10;
    public ImageAdapter(Context c,String[] datos){
        mContext = c;
        datos1=datos;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // TODO Auto-generated method stub
        View v;
        if(convertView==null){
            LayoutInflater li = getLayoutInflater();
            v = li.inflate(R.layout.icon_menu_principal, null);
            TextView tv = (TextView)v.findViewById(R.id.tvicon_text);
            tv.setText(datos1[position].toString());
            ImageView iv = (ImageView)v.findViewById(R.id.icon_image);
            //iv.setImageResource(R.drawable.ic_launcher);
            iv.setImageResource(iconos[position]);
        }
        else
        {
            v = convertView;
        }
        return v;
    }
    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return 6;
    }
    @Override
    public Object getItem(int arg0) {
        // TODO Auto-generated method stub

```

```

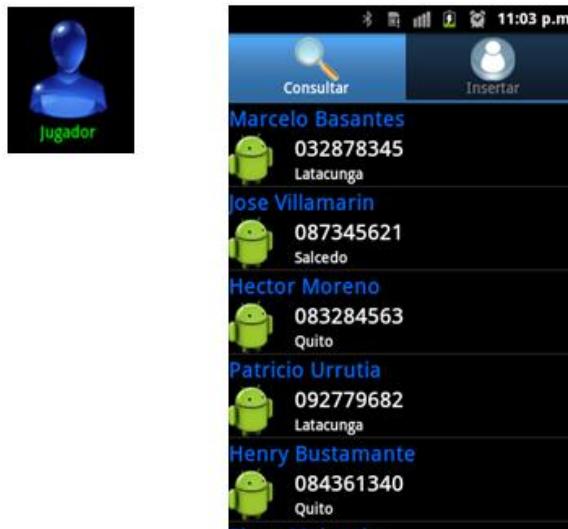
        return null;
    }
    @Override
    public long getItemId(int arg0) {
        // TODO Auto-generated method stub
        return 0;
    }
}

```

Si se elijio el icono **Jugador** esta contiene dos opciones de acceso:

Si elijio la opción consultar le mostrara una lista de jugadores existentes a continuación el codigo generado:

GRAFICO N° 3.47. Layout (jugador_consultar.xml)



REALIZADO POR: La Autora

JugadorConsultar.java

```

package may.android.proyectogrado;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.ListActivity;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ListView;

```

```

public class JugadorConsultar extends ListActivity {
    private RowDatosAdapter adapter;
    String nom;
    String dir;
    String tel;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.jugador_consultar);

        registerForContextMenu(getListView());
        adapter = new RowDatosAdapter(this,DataFramework.getInstance().getEntityList("jugador"));
        setListAdapter(adapter);
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
        // TODO Auto-generated method stub
        super.onCreateContextMenu(menu, v, menuInfo);
        menu.setHeaderTitle("Elija su opción:");
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_ctx_lista, menu);
    }
    @Override
    protected void onItemClick(ListView l, View v, int position, long id) {
        // TODO Auto-generated method stub
        super.onItemClick(l, v, position, id);
        Toast.makeText(this, "Id: " + adapter.getItem(position).getId(), Toast.LENGTH_LONG).show();
    }
    @Override
    public boolean onContextItemSelected(MenuItem item) {
        // TODO Auto-generated method stub
        AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
        switch (item.getItemId()) {
            case R.id.ctxlstEditar:
                Intent intento=new Intent(JugadorConsultar.this,JugadorModificar.class);
                long pos=adapter.getItem(info.position).getId();
                intento.putExtra("par_id", pos);
                intento.putExtra("par_nom",adapter.getItem(info.position).getValue("nombre").toString());
                intento.putExtra("par_dir",adapter.getItem(info.position).getValue("direccion").toString());
                intento.putExtra("par_tel",adapter.getItem(info.position).getValue("telefono").toString());
                startActivity(intento);
                finish();
                return true;
            case R.id.ctxlstEliminar:
                //deleteNote(info.id);
                long posi=adapter.getItem(info.position).getId();
                Toast.makeText(getApplicationContext(),Long.toString(posi),Toast.LENGTH_SHORT).show();
                Entity ent = new Entity("jugador", (long) posi);
                ent.delete();
                finish();
                Intent intento3=new Intent(JugadorConsultar.this,ContenidoTabs.class);
                startActivity(intento3);
                return true;
            default:

```

```

        return super.onContextItemSelected(item);
    }
}
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}
}
}

```

El código anterior permite que al tener presionado sobre el jugador aparezca un cuadro de editar o eliminar jugador

Si selecciona eliminar jugador lo elimina automáticamente.

Si se selecciona editar saldrá la siguiente pantalla a continuación el código generado:

GRAFICO N° 3.48. Layout (jugador_modificar.xml)



REALIZADO POR: La Autora

JugadorModificar.java

```

package may.android.proyectogrado;
import com.android.dataframework.DataFramework;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;
public class JugadorModificar extends Activity {

```

```

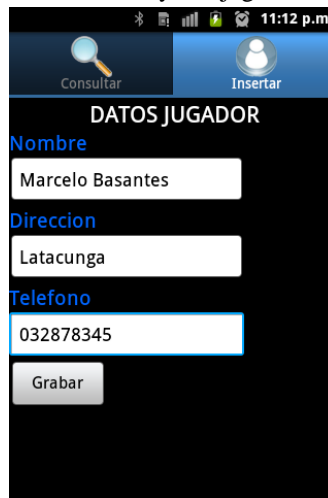
TextView tvparam;
EditText etnom;
EditText etdir;
EditText ettel;
Button btmodificar;
int paramid;
long pos;
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    try {
        DataFramework.getInstance().open(this, "may.android.proyectogrado");
    } catch (Exception e) {
        e.printStackTrace();
    }
    setContentView(R.layout.jugador_modificar);

    pos=getIntent().getExtras().getLong("par_id");
    String parametro_recibido_nombre=getIntent().getExtras().getString("par_nom");
    String parametro_recibido_direccion=getIntent().getExtras().getString("par_dir");
    String parametro_recibido_telefono=getIntent().getExtras().getString("par_tel");
    tvparam=(TextView) findViewById(R.id.txtparam);
    tvparam.setText(parametro_recibido_nombre);
    etnom=(EditText) findViewById(R.id.etmodnombre);
    etdir=(EditText) findViewById(R.id.etmoddireccion);
    ettel=(EditText) findViewById(R.id.etmodtelefono);
    btmodificar=(Button) findViewById(R.id.btnmodificar);
    etnom.setText(parametro_recibido_nombre);
    etdir.setText(parametro_recibido_direccion);
    ettel.setText(parametro_recibido_telefono);
    btmodificar.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Entity ent = new Entity("jugador", (long) pos);
            ent.setValue("nombre", etnom.getText().toString());
            ent.setValue("direccion", etdir.getText().toString());
            ent.setValue("telefono", ettel.getText().toString());
            ent.save();
            finish();
            Intent intento=new Intent(JugadorModificar.this, ContenidoTabs.class);
            startActivity(intento);
        }
    });
}
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}
}

```

Si elijio la opción insertar le pedirá que ingrese el nombre, dirección, teléfono, continuación el codigo generado:

GRAFICO N° 3.49. Layout (jugador_insertar.xml)



REALIZADO POR: La Autora

JugadorInsertar.java

```
package may.android.proyectogrado;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View.OnClickListener;
import android.widget.EditText;
public class JugadorInsertar extends Activity{
    EditText etnomju;
    EditText etdirju;
    EditText ettelju;
    Button boton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.jugador_insertar);

        etnomju=(EditText) findViewById(R.id.etnomjug);
        etdirju=(EditText) findViewById(R.id.etdirjug);
        ettelju=(EditText) findViewById(R.id.etteljug);
        boton=(Button) findViewById(R.id.btngrabarjug);
        boton.setOnClickListener(new OnClickListener() {
```

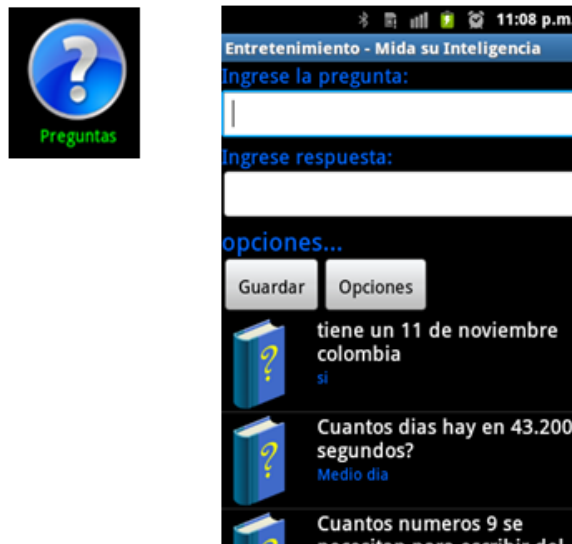
```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    Entity ent = new Entity("jugador");
    ent.setValue("nombre", etnomju.getText().toString());
    ent.setValue("direccion", etdirju.getText().toString());
    ent.setValue("telefono", ettelju.getText().toString());
    ent.save();
    finish();
    Intent intento=new Intent(JugadorInsertar.this,ContenidoTabs.class);
    startActivity(intento);
}
});
}
@Override
protected void onDestroy() {
    super.onDestroy();
    DataFramework.getInstance().close();
}
}

```

Si elijio el icono **Preguntas** debe ingresar la pregunta y la opción de respuesta cada una se ira almacenando en una lista para su verificación de ingreso, a continuación el codigo generado.

GRAFICO N° 3.50. Layout (preguntas.xml)



REALIZADO POR: La Autora

Preguntas.java

```
package may.android.proyectogrado;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.os.Bundle;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class Preguntas extends Activity {
    private RowDatosAdapterPreguntas adapter;
    Button boton;
    Button botonagregar;
    Button botonactualizar;
    ListView listapreguntas;
    EditText etp;
    EditText etr;
    long idpreg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.preguntas);

        boton=(Button) findViewById(R.id.btIrAgregarOpciones);
        botonagregar=(Button) findViewById(R.id.btAgregar);
        botonactualizar=(Button) findViewById(R.id.btActualizarPregunta);
        listapreguntas=(ListView) findViewById(R.id.lstPreguntas);
        etp=(EditText) findViewById(R.id.etpregunta);
        etr=(EditText) findViewById(R.id.etrespuesta);
        consultar();
        registerForContextMenu(listapreguntas);
        boton.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Intent intento=new Intent(Preguntas.this,OpcionesPregunta.class);
                startActivity(intento);
            }
        });
        botonagregar.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Entity ent = new Entity("bancopreguntas");
                ent.setValue("pregunta", etp.getText().toString());
                ent.setValue("respuesta", etr.getText().toString());
                ent.save();
                etp.setText("");
                etr.setText("");
            }
        });
    }
}
```

```

        etp.setFocusable(true);
        consultar();
    }
});
botonactualizar.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        botonagregar.setVisibility(View.VISIBLE);
        botonactualizar.setVisibility(View.INVISIBLE);
        Entity entact = new Entity("bancopreguntas", (long) idpreg);
        entact.setValue("pregunta", etp.getText().toString());
        entact.setValue("respuesta", etr.getText().toString());
        entact.save();
        finish();
        Intent intentoact=new Intent(Preguntas.this,Preguntas.class);
        startActivity(intentoact);
    }
});
}
@Override
public void onCreateContextMenu(ContextMenu menu, View v,ContextMenu.ContextMenuInfo
menuInfo) {
    // TODO Auto-generated method stub
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.setHeaderTitle("Elija su opción:");
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_ctx_lista, menu);
}
public boolean onOptionsItemSelected(MenuItem item) {
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.ctxlstEditar:
            botonactualizar.setVisibility(View.VISIBLE);
            botonagregar.setVisibility(View.INVISIBLE);
            idpreg=adapter.getItem(info.position).getId();
            etp.setText(adapter.getItem(info.position).getValue("pregunta").toString());
            etr.setText(adapter.getItem(info.position).getValue("respuesta").toString());
            return true;
        case R.id.ctxlstEliminar:
            long posi=adapter.getItem(info.position).getId();
            Toast.makeText(getApplicationContext(),Long.toString(posi),Toast.LENGTH_SHORT).show();
            Entity ent = new Entity("bancopreguntas", (long) posi);
            ent.delete();
            finish();
            Intent intento3=new Intent(Preguntas.this,Preguntas.class);
            startActivity(intento3);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}
public void consultar(){

```

```

        adapter = new
RowDatosAdapterPreguntas(this,DataFramework.getInstance().getEntityList("bancopreguntas"));
        listapreguntas.setAdapter(adapter);
    }
}

```

El código anterior permite que al tener presionado sobre la pregunta ingresada aparezca un cuadro de editar o eliminar la pregunta

Si selecciona eliminar pregunta la elimina automáticamente.

Si se selecciona editar saldrá la siguiente pantalla a continuación el código generado:

GRAFICO N° 3.51. Layout (preguntas_listado.xml)



REALIZADO POR: La Autora

PreguntasListado.java

```

package may.android.proyectogrado;
import java.util.List;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ListView;
import android.widget.Toast;
public class PreguntasListado extends Activity {
    ListView listadopreguntas;
    RowDatosAdapterPreguntasListado adapter;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    try {
        DataFramework.getInstance().open(this, "may.android.proyectogrado");
    } catch (Exception e) {
        e.printStackTrace();
    }
    setContentView(R.layout.preguntas_listado);

    listadopreguntas=(ListView) findViewById(R.id.lstlistadopreguntas);
    consultar();
    listadopreguntas.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int pos,long arg3) {
            // TODO Auto-generated method stub
            Toast.makeText(getApplicationContext(), "Id: " + adapter.getItem(pos).getId(),
            Toast.LENGTH_LONG).show();
            consultaropcionespregunta(adapter.getItem(pos).getId());
        }
    });
}

public void consultaropcionespregunta(long idpreg){
    Toast.makeText(getApplicationContext(), "entroalmetodo", Toast.LENGTH_SHORT).show();
    List<Entity> opciones = DataFramework.getInstance().getEntityList("preguntaopciones",
    "bancopreguntas_id = "+idpreg, "");
    Iterator iter = opciones.iterator();
    while (iter.hasNext()) {
        Entity ent = (Entity)iter.next();
        System.out.println(ent.getString("opcion"));
        Toast.makeText(getApplicationContext(), ent.getString("opcion").toString(),
        Toast.LENGTH_SHORT).show();
    }
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}

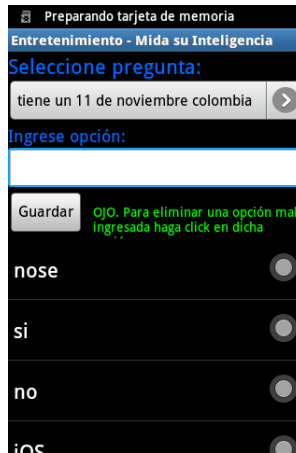
public void consultar(){
    adapter = new
    RowDatosAdapterPreguntasListado(this,DataFramework.getInstance().getEntityList("bancopreguntas"));
    listadopreguntas.setAdapter(adapter);
}
}

```

Al dar clic sobre el botón opciones aparecerá la layout para ingresar las opciones a cada pregunta:

Escogerá la pregunta que desea ingresar las opciones y por cada ingreso de opción dar clic en guardar:

GRAFICO N° 3.52. Layout(opciones_pregunta.xml)



REALIZADO POR: La Autora

OpcionesPregunta.java

```
package may.android.proyectogrado;
import java.util.List;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View.OnClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;
public class OpcionesPregunta extends Activity {
    private List<Entity> preguntas;
    Spinner sppreg;
    SimpleCursorAdapter adapter;
    ArrayAdapter<String> adaptadoropcion;
    List<Entity> opciones;
    Button boton;
    EditText opcion;
    String id;
    String name;
    ListView lvOpciones;
    String[] arrayOpciones;
    long[] arrayIdOpciones;
    long idOpcion;
}
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    try {
        DataFramework.getInstance().open(this, "may.android.proyectogrado");
    } catch (Exception e) {
        e.printStackTrace();
    }
    setContentView(R.layout.opciones_pregunta);

    sppreg=(Spinner) findViewById(R.id.spPreguntas);
    lvOpciones=(ListView) findViewById(R.id.lstOpcionesGuardar);
    boton=(Button) findViewById(R.id.btAgregarOpcion);
    opcion=(EditText) findViewById(R.id.etopcion);
    Cursor c=DataFramework.getInstance().getEntityCursor("bancopreguntas","", "");
    adapter = new SimpleCursorAdapter(this,android.R.layout.simple_spinner_item,c,new
    String[] {"pregunta","_id"}, new int[] {android.R.id.text1});
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    sppreg.setAdapter(adapter);
    sppreg.setOnItemClickListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> arg0, View arg1,int pos, long arg3) {
            // TODO Auto-generated method stub
            Cursor cursor = (Cursor) sppreg.getSelectedItem();
            id = cursor.getString(0);
            name = cursor.getString(1);
            Toast.makeText(getApplicationContext(), id+" - "+name,Toast.LENGTH_SHORT).show();
            ConsultarOpcionesPregunta((Long.parseLong(id)));
        }

        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub
        }
    });
    boton.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Entity ent = new Entity("preguntaopciones");
            ent.setValue("opcion",opcion.getText().toString());
            ent.setValue("bancopreguntas_id", id);
            ent.save();
            ConsultarOpcionesPregunta(Long.parseLong(id));
        }
    });
    lvOpciones.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int pos,long arg3) {
            // TODO Auto-generated method stub
            idOpcion=arrayIdOpciones[pos];
            Toast.makeText(getApplicationContext(),"idOpcion="+idOpcion,Toast.LENGTH_SHORT).show();
            showDialog(0);
        }
    });
}

```

```

@Override
protected Dialog onCreateDialog(int id){
    Dialog dialogo=crearAlertaConfirmarCierreAplicacion(); //Falta implementar esta función
    return dialogo;
}

private AlertDialog crearAlertaConfirmarCierreAplicacion() {
    AlertDialog miAlerta=null;
    AlertDialog.Builder miConstructor=new AlertDialog.Builder(this);
    String cadResp="Desea eliminar la opción?";
    miConstructor.setTitle(cadResp);
    miConstructor.setIcon(R.drawable.eliminar);
    miConstructor.setCancelable(true);
    miConstructor.setPositiveButton(R.string.si, new DialogInterface.OnClickListener(){
        public void onClick(DialogInterface dialog, int which){
            Entity ent = new Entity("preguntaopciones", (long) idOpcion);
            ent.delete();
            ConsultarOpcionesPregunta(Long.parseLong(id));
        }
    });
    miConstructor.setNegativeButton(R.string.no, new DialogInterface.OnClickListener(){
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    miAlerta=miConstructor.create();
    return miAlerta;
}

public void ConsultarOpcionesPregunta(long idpreg){
    opciones = DataFramework.getInstance().getEntityList("preguntaopciones",
"bancopreguntas_id = "+idpreg, "");
    Iterator iter = opciones.iterator();
    arrayOpciones=new String[opciones.size()];
    arrayIdOpciones=new long[opciones.size()];
    int i=0;
    while (iter.hasNext()) {
        Entity ent1 = (Entity)iter.next();
        arrayOpciones[i]=ent1.getString("opcion").toString();
        arrayIdOpciones[i]=ent1.getId();
        i++;
    }
    llenarlistaopcionescontestar();
}

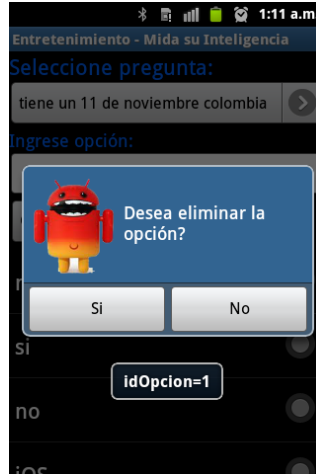
public void llenarlistaopcionescontestar(){
    adaptadoropcion = new
    ArrayAdapter<String>(this,android.R.layout.simple_list_item_single_choice,
    android.R.id.text1, arrayOpciones);
    lvOpciones.setAdapter(adaptadoropcion);
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}
}

```

Si se llego a ingresar mal las opciones al dar clic sobre la opción mal ingresada se puede eliminar y volverla a ingresar

GRAFICO N° 3.53. Layout(eliminar opcion)



REALIZADO POR: La Autora

Si elijio el icono **Jugar** empezara el juego de mida su inteligencia previo a esto le preguntara si ha leído las instrucciones de uso de juego, a continuación el codigo generado:

GRAFICO N° 3.54. Layout (jugar_layout2)



REALIZADO POR: La Autora

JugadorLayuot2.java

```
package may.android.proyectogrado;
import java.util.Iterator;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.SimpleCursorAdapter;
import android.widget.Spinner;
import android.widget.Toast;
public class JugadorLayuot2 extends Activity {
    Spinner comboJugadores;
    SimpleCursorAdapter adapter;
    Button botonContinuar;
    TextView cajaPregunta,cajaNumeroPregunta,cajaNumBuenas,cajaNumMalas;
    ListView listacontestar;
    int numeroAleatorio;
    int[] arregloIdPreguntas=new int[5];
    long[] arregloTodasIdPreguntas;
    int idpreg=-1,m=0,numeroBuenas=0,numeroMalas=0;
    int idJugador=-1;
    String respuestaCorrecta="";
    String[] arrayOpciones;
    String nombreJugador=null;
    long idUltimaPartida=-1;
    String[] rc=new String[5];
    String[] rv=new String[5];
    String[] acc=new String[5];
    String resul="";
    int ii=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.jugar_layout2);

        comboJugadores=(Spinner) findViewById(R.id.spJugadores);
        botonContinuar=(Button) findViewById(R.id.btnContinuar);
        cajaPregunta=(TextView) findViewById(R.id.tvMuestraPregunta);
        cajaNumeroPregunta=(TextView) findViewById(R.id.textView2);
        listacontestar=(ListView) findViewById(R.id.lvOpcionesContestar);
        cajaNumBuenas=(TextView) findViewById(R.id.tvCantBuenas);
        cajaNumMalas=(TextView) findViewById(R.id.tvCantMalas);
        ConsultarJugadores();
        GenerarIds();
        comboJugadores.setOnItemClickListener(new OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> arg0, View arg1,
                int pos, long arg3) {
                // TODO Auto-generated method stub
```

```

        String spinnerString = null;
        Cursor cc = (Cursor)(comboJugadores.getSelectedItemId());
        if (cc != null) {
            spinnerString = cc.getString(cc.getColumnIndex("nombre"));
        }
        nombreJugador=spinnerString;
        idJugador=(int) comboJugadores.getSelectedItemId();
String datosjugador=spinnerString + ": "+comboJugadores.getSelectedItemId();
Toast.makeText(arg0.getContext(),datosjugador,Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub
    }
});
botonContinuar.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        try{
            idpreg=arregloIdPreguntas[m];
            Entity ent = new Entity("bancopreguntas", (long)idpreg);
            String pregunta=ent.getString("pregunta");
            respuestaCorrecta=ent.getString("respuesta");
            cajaPregunta.setText(arregloIdPreguntas[m]+" "+pregunta);
            cajaNumeroPregunta.setText("Pregunta No. "+(m+1));
            ConsultarOpcionesPregunta(idpreg);
        }catch(Exception ex){
        }
    }
});
listacontestar.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int pos, long arg3) {
        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), listacontestar.getItemAtPosition(pos).toString(),
        Toast.LENGTH_SHORT).show();
        String respuestaContestada=listacontestar.getItemAtPosition(pos).toString();
        String ac;
        if(respuestaCorrecta.equals(respuestaContestada))
        {
            numeroBuenas++;
            ac="Acerto";
        }
        else
        {
            numeroMalas++;
            ac="No acerto";
        }
        rc[ii]=respuestaContestada;
        rv[ii]=respuestaCorrecta;
        acc[iii]=ac;
        cajaNumBuenas.setText("Buenas: "+numeroBuenas);
        cajaNumMalas.setText("Malas: "+numeroMalas);
        if(m+1==5)
        {
            AsignarResultado();
            GrabarPartida();
            showDialog(0);
        }
    }
});

```

```

        m++;
        ii++;
    }
});
}
public void AsignarResultado(){
    if(numeroBuenas==0)
        resul="Malo";
    if(numeroBuenas>=1 && numeroBuenas<=2)
        resul="Bueno";
    if(numeroBuenas>=3 && numeroBuenas<=4)
        resul="Muy Bueno";
    if(numeroBuenas==5)
        resul="Excelente";
}
protected Dialog onCreateDialog(int id){
    Dialog dialogo=crearDialogoAlerta(); //Falta implementar esta función
return dialogo;
}
private Dialog crearDialogoAlerta()
{
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setIcon(R.drawable.dialog);
    builder.setTitle("Informacion");
    String cadResp="Buenas: "+numeroBuenas+"\nMalas: "+numeroMalas+"\nResultado: "+resul
    +"\n!Salir de partida.!";
    builder.setMessage(cadResp);
    builder.setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            JugadorLayuot2.this.finish();
        }
    });
    return builder.create();
}
public void GrabarPartida(){
    Toast.makeText(getApplicationContext(), "Ingreso partida:"+Integer.toString(idJugador),
    Toast.LENGTH_SHORT).show();
    Date fechasistema=new Date();
    String fa=fechasistema.getDate()+"
    "+ObtenerMes(fechasistema.getMonth()+1)+"."+fechasistema.getYear()+1900);
    String ha=fechasistema.getHours()+":"+fechasistema.getMinutes()+":"+fechasistema.getSeconds();
    Entity ent = new Entity("partida");
    ent.setValue("fechapar", fa);
    ent.setValue("horapar", ha);
    ent.setValue("jugador_id",idJugador);
    ent.save();
    ObtenerIdUltimaPartida();
    GrabarPreguntasContestadas();
    GrabarResumenPartida();
}
public String ObtenerMes(int numes){
    String[] meses={"Ene","Feb","Mrz","Abr","May","Jun","Jul","Ago","Spt","Oct","Nov","Dic"};
    return meses[numes-1].toString();
}
public void ObtenerIdUltimaPartida(){
    List<Entity> opciones = DataFramework.getInstance().getEntityList("partida","","");
    Iterator iter = opciones.iterator();
    while (iter.hasNext()) {
        Entity ent1 = (Entity)iter.next();
        idUltimaPartida=ent1.getId();
    }
}

```

```

Toast.makeText(getApplicationContext(), "Ultima partida:"+idUltimaPartida, Toast.LENGTH_SHORT).show();
}
public void GrabarPreguntasContestadas(){
    for(int i=0;i<arregloIdPreguntas.length;i++)
    {
        Entity ent = new Entity("preguntascontestadas");
        ent.setValue("respuesta", rc[i]);
        ent.setValue("partida_id", idUltimaPartida);
        ent.setValue("bancopreguntas_id", arregloIdPreguntas[i]);
        ent.setValue("respuestaverdadera", rv[i]);
        ent.setValue("accion", acc[i]);
        ent.save();
    }
}
public void GrabarResumenPartida(){
    Entity ent = new Entity("resumenpartida");
    ent.setValue("buenas", numeroBuenas);
    ent.setValue("malas", numeroMalas);
    ent.setValue("resultado", resul);
    ent.setValue("partida_id", idUltimaPartida);
    ent.save();
}
public void ConsultarOpcionesPregunta(long idpreg){
    List<Entity> opciones = DataFramework.getInstance().getEntityList("preguntaopciones",
    "bancopreguntas_id = "+idpreg, "");
    Iterator iter = opciones.iterator();
    arrayOpciones=new String[opciones.size()];
    int i=0;
    while (iter.hasNext()) {
        Entity ent1 = (Entity)iter.next();
        arrayOpciones[i]=ent1.getString("opcion").toString();
        i++;
    }
}
llenarlistaopcionescontestar();
}
public void llenarlistaopcionescontestar(){
    ArrayAdapter<String> adaptador = new
ArrayAdapter<String>(this, android.R.layout.simple_list_item_single_choice, android.R.id.text1, arrayOpciones);
    listacontestar.setAdapter(adaptador);
}
public void GenerarIds(){
    ConsultarTodasLasIdsDePreguntas();
    int j=0;
    int sw;
    try{
        while(j<5){
            numeroAleatorio=(int) (Math.random()*arregloTodasIdPreguntas.length-1);
            if (numeroAleatorio==0)
                numeroAleatorio++;
            sw=0;
            for(int k=0;k<arregloIdPreguntas.length;k++){
                if(arregloIdPreguntas[k]==arregloTodasIdPreguntas[numeroAleatorio])
                    {sw=1; break;}
            }
            if(sw==0)
            {
                arregloIdPreguntas[j]=(int) arregloTodasIdPreguntas[numeroAleatorio];
                j++;
            }
        }
    }
}
}

```

```

        } catch (Exception ex) {
        }
    }
    public void ConsultarTodasLasIdsDePreguntas() {
        List<Entity> opciones = DataFramework.getInstance().getEntityList("bancopreguntas", "", "");
        Iterator iter = opciones.iterator();
        arregloTodasIdPreguntas = new long[opciones.size()];
        int i = 0;
        while (iter.hasNext()) {
            Entity ent1 = (Entity) iter.next();
            arregloTodasIdPreguntas[i] = ent1.getId();
            i++;
        }
    }
    public void ConsultarJugadores() {
        Cursor c = DataFramework.getInstance().getEntityCursor("jugador", "", "");
        adapter = new SimpleCursorAdapter(this, android.R.layout.simple_spinner_item, c, new
String[] { "nombre" }, new int[] { android.R.id.text1 });
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        comboJugadores.setAdapter(adapter);
    }
    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        super.onDestroy();
        DataFramework.getInstance().close();
    }
}

```



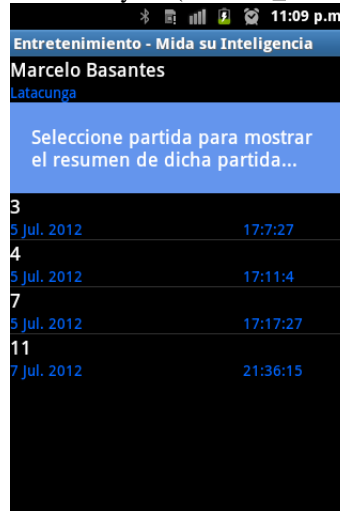
```

    ObtenerJugadores();
    lvJugadores.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int pos,long arg3) {
            Toast.makeText(getApplicationContext(), "-:" +adapter.getItem(pos).getId(),Toast.LENGTH_SHORT).show();
            Intent intento=new Intent(ResumenListadoJugadores.this,ResumenListadoPartidas.class);
            long idju=(long)adapter.getItem(pos).getId();
            intento.putExtra("idjugador",idju);
            intento.putExtra("par_nom",adapter.getItem(pos).getValue("nombre").toString());
            intento.putExtra("par_dir",adapter.getItem(pos).getValue("direccion").toString());
            startActivity(intento);
        }
    });
    lvJugadores.setOnItemClickListener(new OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> arg0, View arg1,int pos, long arg3) {
            // TODO Auto-generated method stub
            Toast.makeText(getApplicationContext(), "-:"
            "+adapter.getItem(pos).getId(),Toast.LENGTH_SHORT).show();
            Intent intento=new
            Intent(ResumenListadoJugadores.this,ResumenListadoPartidas.class);
            long idju=(long)adapter.getItem(pos).getId();
            intento.putExtra("idjugador",idju);
            intento.putExtra("par_nom",adapter.getItem(pos).getValue("nombre").toString());
            intento.putExtra("par_dir",adapter.getItem(pos).getValue("direccion").toString());
            startActivity(intento);
        }
        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub
        }
    });
}
public void ObtenerJugadores(){
    adapter = new
    RowResumenListadoJugadores(this,DataFramework.getInstance().getEntityList("jugador"));
    lvJugadores.setAdapter(adapter);
}
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    super.onDestroy();
    DataFramework.getInstance().close();
}
}

```

Al dar clic sobre el jugador elegido anteriormente aparecerá la siguiente layout mostrando la cantidad de partidas obtenidas por el jugador

GRAFICO N° 3.56. Layout (resumen_listado_partidas.xml)



REALIZADO POR: La Autora

ResumenListadoPartidas.java

```

package may.android.proyectogrado;
import java.util.List;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.TextView;
public class ResumenListadoPartidas extends Activity{
    RowResumenListadoPartidas adapter;
    TextView tvnom, tvdir;
    ListView lvPartidas;
    long pa;
    ClasePartidas[] objPartidas;
    int numBuenas, numMalas;
    String resultado;
    long numPartida;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.resumen_listado_partidas);
    }
}

```

```

pa=getIntent().getExtras().getLong("idjugador");
String parametro_recibido_nombre=getIntent().getExtras().getString("par_nom");
String parametro_recibido_direccion=getIntent().getExtras().getString("par_dir");
tvnom=(TextView) findViewById(R.id.tvnomresumpartida);
tvdir=(TextView) findViewById(R.id.tvdirresumpartida);
tvnom.setText(parametro_recibido_nombre);
tvdir.setText(parametro_recibido_direccion);
lvPartidas=(ListView) findViewById(R.id.listaVisibleListadoPartidas);
ConsultarPartidas(pa);
adapter=new RowResumenListadoPartidas(this,objPartidas);
lvPartidas.setAdapter(adapter);
lvPartidas.setOnItemClickListener(new OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int pos,long arg3) {
// TODO Auto-generated method stub
ConsultarResumen(Long.parseLong(objPartidas[pos].getIdPa()));
}
});
}
public void ConsultarResumen(long idpar){
List<Entity> opciones = DataFramework.getInstance().getEntityList("resumenpartida",
"partida_id="+idpar, "");

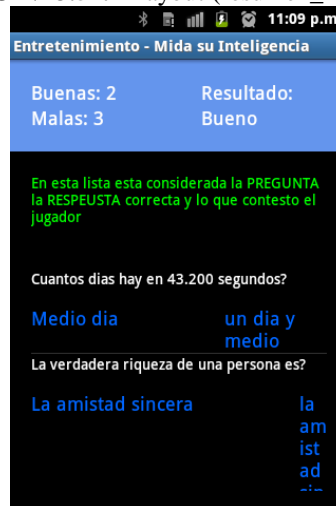
Iterator iter = opciones.iterator();
numPartida=idpar;
while (iter.hasNext()) {
Entity ent1 = (Entity)iter.next();
numBuenas=ent1.getInt("buenas");
numMalas=ent1.getInt("malas");
resultado=ent1.getString("resultado");
}
Intent intento=new Intent(ResumenListadoPartidas.this, ResumenResumen.class);
intento.putExtra("param_nb", numBuenas);
intento.putExtra("param_nm", numMalas);
intento.putExtra("param_resul",resultado);
intento.putExtra("param_np", numPartida);
startActivity(intento);
}
public void ConsultarPartidas(long idjug){
List<Entity> opciones = DataFramework.getInstance().getEntityList("partida", "jugador_id =
"+idjug, "");

Iterator iter = opciones.iterator();
objPartidas=new ClasePartidas[opciones.size()];
int i=0;
while (iter.hasNext()) {
Entity ent1 = (Entity)iter.next();
objPartidas[i]=new
ClasePartidas(Long.toString(ent1.getId()).toString(),ent1.getString("fechapar").toString(),
ent1.getString("horapar"));
i++;
}
}
@Override
protected void onDestroy() {
// TODO Auto-generated method stub
super.onDestroy();
super.onDestroy();
DataFramework.getInstance().close();
}
}

```

Al dar clic sobre la partida de cual se desea saber los resultados mostrara la siguiente layout indicando detalles de la partida:

GRAFICO N° 3.57. Layout (resumen_resumen.xml)



REALIZADO POR: La Autora

ResumenResumen.java

```

package may.android.proyectogrado;
import java.util.List;
import com.android.dataframework.DataFramework;
import com.android.dataframework.Entity;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class ResumenResumen extends Activity{
    int numBuenas,numMalas;
    long idPartida;
    String resultado;
    RowResumenPreguntas adapter;
    TextView tvB,tvM,tvR,tvP;
    ListView lvRP;
    ClaseResumenPreguntas[] resumenpreguntas;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        try {
            DataFramework.getInstance().open(this, "may.android.proyectogrado");
        } catch (Exception e) {
            e.printStackTrace();
        }
        setContentView(R.layout.resumen_resumen);

        numBuenas=getIntent().getExtras().getInt("param_nb");
        numMalas=getIntent().getExtras().getInt("param_nm");
    }
}

```

```

        resultado=getIntent().getExtras().getString("param_resul");
        idPartida=getIntent().getExtras().getLong("param_np");
        Toast.makeText(getApplicationContext(), "Entro id partida es: "+idPartida,
Toast.LENGTH_SHORT).show();
        tvB=(TextView) findViewById(R.id.txtBuenas);
        tvM=(TextView) findViewById(R.id.txtMalas);
        tvR=(TextView) findViewById(R.id.txtResultado);
        tvB.setText("Buenas: "+numBuenas);
        tvM.setText("Malas: "+numMalas);
        tvR.setText(resultado);
        ConsultarPreguntasRespuestasContesto(idPartida);
        lvRP=(ListView) findViewById(R.id.lvRespuestasContesto);
        adapter=new RowResumenPreguntas(this,resumenpreguntas);
        lvRP.setAdapter(adapter);
    }
    public void ConsultarPreguntasRespuestasContesto(long idpa){
        List<Entity> opciones = DataFramework.getInstance().getEntityList("preguntascontestadas",
"partida_id="+idpa,"");
        String pregunta="";
        resumenpreguntas=new ClaseResumenPreguntas[opciones.size()];
        Iterator iter = opciones.iterator();
        int i=0;
        while (iter.hasNext()) {
            Entity ent1 = (Entity)iter.next();
            pregunta=ObtenerPregunta(ent1.getLong("bancopreguntas_id"));
            resumenpreguntas[i]=new
ClaseResumenPreguntas(pregunta,ent1.getString("respuestaverdadera"),ent1.getString("respuesta"));
            i++;
        }
    }
    public String ObtenerPregunta(long idpreg){
        List<Entity> opciones = DataFramework.getInstance().getEntityList("bancopreguntas",
"_id="+idpreg,"");
        String preguntap1="";
        Iterator iter = opciones.iterator();
        while (iter.hasNext()) {
            Entity ent1 = (Entity)iter.next();
            preguntap1=ent1.getString("pregunta");
        }
        return(preguntap1);
    }
    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        super.onDestroy();
        DataFramework.getInstance().close();
    }
}

```

Si elijio el icono **ACERCA DE** aquí se indica pequeños datos técnicos de autoría de la aplicación, a continuación el código generado:

GRAFICO N° 3.58. Layout (acerca_de.xml)



REALIZADO POR: La Autora

AcercaDe.java

```

package may.android.proyectogrado;

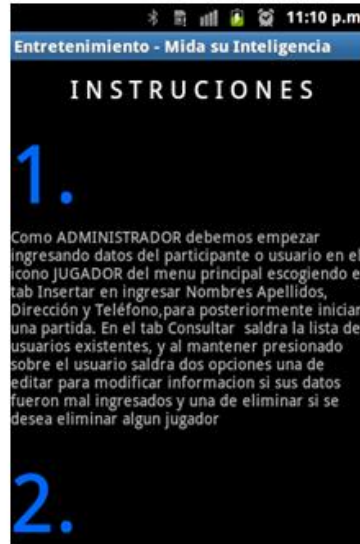
import android.app.Activity;
import android.os.Bundle;
import android.widget.ListView;
public class AcercaDe extends Activity {
    ListView lvAcerca;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.acerca_de);

        lvAcerca=(ListView) findViewById(R.id.lvAcercade);
        final ClaseAcercaDe[] datos =
            new ClaseAcercaDe[]{
                new ClaseAcercaDe("Autor", "Maritza Maigualca"),
                new ClaseAcercaDe("Sistema Operativo", "Android 2.3.4"),
                new ClaseAcercaDe("Tema", "Mida su inteligencia - Entretenimiento"),
                new ClaseAcercaDe("Versión", "1.0"),
                new ClaseAcercaDe("Derechos reservados", "Autor M.E.M.M");
        AdaptadorAcercaDe adaptador = new AdaptadorAcercaDe(this,datos);
        lvAcerca.setAdapter(adaptador);
    }
}

```

El sexto icono pertenece a **Instrucciones** de uso de la aplicación y pasos a seguir para iniciar una partida, a continuación el código generado:

GRAFICO N° 3.59. Layout(instrucciones)



REALIZADO POR: La Autora

Instrucciones.java

```
package may.android.proyectogrado;
```

```
import android.app.Activity;  
import android.os.Bundle;
```

```
public class Instrucciones extends Activity{  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.instrucciones);  
    }  
}
```

3.6.3.3. Pruebas

PLAN DE PRUEBAS (APLICACIÓN MIDA SU INTELIGENCIA)

Todo software debe pasar por un proceso de pruebas, sin lugar a duda esta aplicación de Mida su Inteligencia se encuentra dentro de las categorías de entretenimiento para usuarios de dispositivos móviles (celular) que tengan como software base Android. Esta aplicación esta considerada a manera de un juego como parte investigativa de la postulante que se adentra en el desarrollo de este tipo aplicaciones y que debe considerarse como obligatorio el proceso de pruebas.

Introducción.

El presente documento tiene como objetivo describir las pruebas a las que se debe someter la aplicación de entretenimiento (Mida su inteligencia) en sus diferentes fases de desarrollo a fin de cumplir con los requisitos establecidos en la fase de análisis.

Se pretende que el proceso de plan de pruebas forme parte del desarrollo en sí de manera que la mayoría de módulos desarrollados sean probados una vez codificados, sin necesidad de esperar al final del proyecto. El desarrollo del proyecto, basado en metodología XP, permite la frecuente generación de pequeñas soluciones o módulos la mayoría de los cuales son testeables de forma independiente. Este tipo de proyectos permite a los desarrolladores recibir un feedback rápido por parte del equipo de testing u otros desarrolladores de manera que los errores no se hereden ni se magnifiquen en siguientes versiones cuando el volumen de código es mayor.

Identificación del plan.

- **TP-Req-Aplicación (plan de verificación del requerimiento con el sistema).**- Este plan se aplica asegurando que los requerimientos se hayan implementado en la aplicación y su correcto funcionamiento, desde el punto de vista de administrador de la aplicación.

- **Pruebas funcionales.-** La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:
 - Los procesos y reglas de negocio establecidas.
 - Que se cumplan los requerimientos funcionales establecidos.

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

Alcance.

Se van aplicar dos tipos de pruebas, las cuales determinarán el correcto funcionamiento de la aplicación como así mismo, los correctivos necesarios que se deban aplicar para que la aplicación tenga su aceptación total por parte de los usuarios finales (administrador, jugador).

Actores.

Administrador.- Sera la persona que sea dueño del dispositivo móvil, sera el propietario y a su vez administrador de la aplicación, además podría ser un posible jugador al mismo tiempo.

Jugador.-Sera la persona quien directamente luego de registrado como jugador, sea el que conteste el test de mida su inteligencia.

Desarrollo.

TP-REQ-APLICACIÓN (PLAN DE VERIFICACIÓN DEL REQUERIMIENTOS CON EL SISTEMA)

TABLA N° 3.14. Plan de Verificación del Requerimientos con el Sistema

REQUERIMIENTO	PROGRAMADOR	RESPONSABLES VERIFICACIÓN	IMPLEMENTADO	OBSERVACIONES
R1. En la aplicación debe existir una pantalla inicial de presentación esta contendrá: autor de la aplicación, versión de software y una pequeña introducción.	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	SI	Ninguna
R2. Antes de iniciar el juego la aplicación debe mostrar un menú de opciones para la administración de los diferentes componentes de la aplicación como pueden ser: Jugador, Preguntas, Jugar, Resumen, Acerca de, Otros, sin estar sujetos a estas, ya que pueden existir modificaciones o alteraciones durante el desarrollo de la aplicación.	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	SI	Se debe hacer el cambio de la opción Otros por Instrucciones del juego.
R3. La aplicación debe permitir dentro de la opción JUGADOR , gestionar los datos del mismo, con los procesos de consultar, insertar, modificar y eliminar datos de un jugador.	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	SI	Ninguna
R4. La aplicación permitirá ejecutar dentro una opción PREGUNTAS , gestionar datos de preguntas con su respectiva respuesta, y adicional permitirá ingresar las opciones a contestar para cada pregunta. Los procesos de realizaran para las el ingreso de preguntas, serán: ingresar, editar, eliminar preguntas.	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	SI	La aplicación debe permitir eliminar opciones de preguntas si es que se ingreso mal.

<p>R5. Dentro de la aplicación existirá La opción JUGAR, la cual, deberá permitir tener acceso al juego de mida su inteligencia a través de preguntas seleccionadas randomicamente de un banco de preguntas. El número de preguntas seleccionadas serán cinco, a las que deberá responder el jugador, y al último emitirá un resultado si el jugador es malo, bueno, muy bueno o sobresaliente.</p>	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	<p>SI</p>	<p>Ninguna.</p>
<p>R6. La opción RESUMEN, deberá acceder a un listado de jugadores, con los cuales al dar clic en uno de ellos se desplegará el resumen del juego, considerando que un jugador puede haber realizado varias partidas, y el resumen consistirá en visualizar cuantas preguntas malas y cuantas preguntas buenas generó en el test y su puntaje obtenido o valoración final</p>	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	<p>SI</p>	<p>Ninguna</p>
<p>R7. La opción ACERCA De, deberá visualizar información importante para el jugador que ofrezca un panorama claro del desarrollo de la aplicación.</p>	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	<p>SI</p>	<p>Ninguna</p>
<p>R8. En la opción Instrucciones, se dará a conocer al jugador las instrucciones de juego y lo que debe hacer antes de jugar</p>	<ul style="list-style-type: none"> • Maritza Maigualca 	<ul style="list-style-type: none"> • Maritza Maigualca • Director de Tesis 	<p>SI</p>	<p>Ninguna</p>

ELABORADO POR: La Autora

PRUEBAS FUNCIONALES. Administrador.

TABLA N° 3.15. Pruebas Funcionales. Administrador.

ACTIVIDAD	ADMINISTRADOR	RESPONSABLES	ENTRADAS	SALIDAS	OBSERVACIONES	ESTADO
Verificar gestión de datos de jugador: ingreso, consulta, modificación, eliminación de datos del jugador.	Varios	Maritza Maigualca	Nombres, apellidos, dirección y teléfono del jugador	Inserción satisfactoria Consulta satisfactoria Actualización Satisfactoria Eliminación satisfactoria	Para eliminar y editar datos de jugador primero debemos consultar jugador existente.	Aprobado
Verificar gestión de datos de preguntas	Varios	Maritza Maigualca	Descripción de la pregunta y su respectiva respuesta	Inserción satisfactoria Consulta satisfactoria Actualización Satisfactoria Eliminación satisfactoria	Para eliminar y editar datos de pregunta primero debemos consultar pregunta existente.	Aprobado
Verificar manipulación de opciones de preguntas, las cuales servirán para que visualice y responda el jugador	Varios	Maritza Maigualca	Pregunta	Inserción satisfactoria Eliminación satisfactoria	Ninguna	Aprobado
Verificar el despliegue de resumen de una partida o varias partidas de juego de diferentes jugadores	Varios	Maritza Maigualca	Datos de Jugador Datos de partida	Obtención del resumen de la partida o partidas satisfactorio.	Ninguna	Aprobado
Verificar los parámetros de acerca del desarrollo de la aplicación. Acerca de.	Varios	Maritza Maigualca	Autor Sistema Operativo Versión SO. Otros	Satisfactorio	Ninguna	Aprobado
Verificar instrucciones del juego	Varios	Maritza Maigualca	Instrucciones	Satisfactorio	Separar las instrucciones.	Aprobado

ELABORADO POR: La Autora

PRUEBAS FUNCIONALES. Jugador.

Fecha: _____

TABLA N° 3.16. Pruebas Funcionales. Administrador.

ACTIVIDAD	JUGADOR	RESPONSABLES	ENTRADAS	SALIDAS	OBSERVACIONES	ESTAD O
Verificar juego	Varios	Maritza Maigualca	<ul style="list-style-type: none"> Datos de jugador Listado de 5 preguntas obtenidas randomicamente 	<ul style="list-style-type: none"> Obtención de la preguntas con sus respectivas opciones. Satisfactorio Obtención del número de contestaciones buenas. Satisfactorio Obtención del número de contestaciones malas. Satisfactorio Almacenamiento de la partida y resumen de la partida. Satisfactorio 	Al responder la pregunta, inmediatamente debe blanquearse y/o pasar a la siguiente pregunta.	Aprobado

ELABORADO POR: La Autora

3.6.4. Fase de Puesta en producción - Implementación

En la fase de producción, se requiere hacer pruebas de rendimiento y comprobación del funcionamiento de la aplicación antes de que esta pueda ser entregada al usuario. En esta fase, aun se pueden encontrar cambios con el fin de poder mejorarlos en las próximas entregas, permitiendo así añadir características adicionales a la versión actual.

Después que la primera entrega es producida para el uso del cliente, el grupo de trabajo debe mantener el funcionamiento de la aplicación mientras se realizan nuevas iteraciones.

3.6.5. Fase de Mantenimiento

La fase de mantenimiento, es el estado normal de un proyecto en XP. En esta fase es necesario producir nuevas funcionalidades manteniendo la existente, mantener a la aplicación existente corriendo manejando las altas y bajas en el equipo e incorporar nuevas personas al equipo de trabajo.

3.6.6. Fase Muerte del proyecto

Un proyecto llega a su final cuando el cliente no aporta con más historias de usuario para que el grupo de trabajo las implemente. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. Con esto puede darse por concluido el desarrollo de la aplicación, ya que ha alcanzado toda la funcionalidad que el cliente deseaba. Otra razón para que un proyecto concluya es la falta de recursos por parte del cliente, ocasionando que el proyecto sea abandonado.

3.7. Conclusiones

Luego de haber culminado el presente trabajo de investigación, se concluye que:

- Gracias a la aplicación de entretenimiento como autora se logro obtener nuevas experiencias en el ámbito de la programación orientado a objetos que es como trabaja java, con android para desarrollar aplicaciones móviles.
- El futuro de Linux y especialmente de Android es uno de los más claros y prometedores por sus propias características de inclusión en el desarrollo, seguridad y transparencia; así como en la versatilidad de su modelo económico donde el Gobierno del Ecuador, empresas públicas, universidades, bancos y muchas empresas están apostando a esta plataforma en la nueva era de la información y las tecnologías libres.
- El sistema operativo Android, es una excelente plataforma para desarrollar aplicaciones, por ser software libre sin necesidad de adquirir licencias de programación o instalación evitándonos los altos costos, además porque se puede empezar a programar con conocimientos de java bien sea en Linux o Windows de cualquier versión.
- Se desarrolló una aplicación móvil de calidad ya que se trabajo con un lenguaje como es java, una base de datos relacional y de dominio publico SQLite contenida en el framework de Android, la cual fue de mucha utilidad en el desarrollo ya que permitio gestionar los datos de la aplicación.
- El uso de un framework especializado en una determinada área de programación hace que el proceso sea más fácil y ágil en el desarrollo, lo que implica un ahorro de tiempo, destacando que hay que hacer uso correcto de las instrucciones correspondientes que el framework ofrece.
- Hasta la presente fecha no existe una metodología para el desarrollo de aplicaciones móviles, por lo cual se empleo la Metodología XP que ayudo a llevar un proceso adecuado, destacando que no se empleo toda la metodología en su totalidad, sino que se acoplo según los requerimientos para el desarrollo.

3.8. Recomendaciones

- Se recomienda que al trabajar con nuevas tecnologías en este caso para dispositivos móviles es importante tener la suficiente documentación ya que sirve de apoyo para seguir los pasos correctos y obtener un software (aplicación) de calidad, leer documentos, libros, foros en la web, etc.
- Al empezar a programar sobre Android se debe tener en cuenta los tipos de recursos que se van a emplear durante el proceso de desarrollo, ya que dependiendo de dichos recursos demorará el tiempo de compilación y ejecución de la aplicación, para que así no se detenga por completo el IDE de programación.
- Cuando se hace uso de la base de datos, en lo posible tratar de sincronizar el dispositivo virtual y el dispositivo físico para que no exista un desfase de datos y así lograr obtener los resultados requeridos, sin tener que estar digitando nuevamente los datos para las pruebas en ambos tipos de dispositivos.
- Como investigadora recomiendo a nuestros compañeros de la Carrera en Informática y Sistemas Computacionales inviertan tiempo en la investigación sobre nuevas tecnologías que aparecen en nuestro mercado laboral, ya que muchas de ellas significan una posibilidad sustancial de crecer como profesionales e incluso como empresarios.

3.9. Glosario de Términos

A

- **ADT** (*Android Development Tools*) o : “Plugin de android para eclipse facilita en gran medida el desarrollo de aplicaciones para la plataforma”
- **AMPS** (*Advanced Mobile Phone System*) o Sistema Telefonico Movil Avanzado: “Son sistemas de telefonía móvil de primera generación”
- **API** (*Application Programing Interface*) o Interfaz de Programación de Aplicaciones: “Permite el uso de cierto lenguaje de programación, también incluye hardware sofisticado para comunicarse con un determinado sistema.”
- **Apple**: “Es una empresa multinacional estadounidense que diseña y produce equipos electrónicos y software. Entre los productos de hardware más conocidos de la empresa se cuenta con equipos Macintosh, iPod, iPhone, eliPad. Entre el software de Apple se encuentran el sistema operativo Mac OS X, el sistema operativo iOS, el explorador de contenido multimedia iTunes,”
- **AVD** (*Android Virtual Device*) o Dispositivo Virtual de Android: “Configura las distintas versiones de Android”

B

- **Base de datos**: “Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación”
- **Bugs** (*insecto*): “Un bug es un error o defecto en el software o hardware que hace que un programa funcione incorrectamente. Los bugs son causados por conflictos del software cuando las aplicaciones intentan funcionar en multitarea.”
- **BYTECODE**: “Es el código intermedio entre el código fuente y el código de maquina”

C

- **CDMA** (*Code Division Multiple Access*) o Acceso múltiple por división de código: “Es una red de telefonía celular sin chip, suele utilizarse para referirse a una interfaz de aire inalámbrica de telefonía móvil desarrollada por la empresa Qualcomm”
- **Compilador**: “Es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de máquina (0,1), también puede ser un código intermedio (bytecode), o simplemente texto.”

D

- **Dataframework**: “Esta librería permite gestionar una base de datos SQLite dentro de Android de una manera sencilla.”
- **Dalvik**: “Es la máquina virtual que utiliza la plataforma para dispositivos móviles Android.”
- **.dex** (*Dalvik Executables Format*) o Formato Ejecutable Dalvik: “La herramienta dx incluida en el SDK de Android permite transformar los archivos .Class de Java compilados al formato de archivos .Dex”
- **Debugger** (Depurador): “Es un programa usado para probar y depurar (eliminar los errores) de otros programas”

E

- **Emuladores**: “Es un software que permite ejecutar programas o videojuegos en una plataforma diferente de aquella para la cual fueron escritos originalmente.”
- **Eclipse**: “Es un Entorno de Desarrollo Integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido.”
- **ECJ** (*Compilador de java para eclipse*): “Compilador de java que sirve para compilar proyectos realizados en eclipse ”

F

- **FDMA** (*Frequency Divison Multiple Access*) o Acceso Múltiple por División de Frecuencia: “Es una técnica usada en múltiples protocolos de comunicacion, tanto digitales como analógicos, entre ellos teléfonos móviles de redes GSM.”
- **Framework**: “Los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución)”

G

- **GSM** (*Global System for Mobile Communications*): “Sistema Global para Comunicaciones Mviles”;
- **GUI** (*Graphical User Interface*) o Interfaz Grafica de Usuario: “Es un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles”

H

- **HTML** (*HyperText Markup Language*) o Lenguaje de marcas de hipertexto: “Este lenguaje elabora páginas web”

I

- **IDE** (*Integrated Development Environment*) o Entorno de Desarrollo Integrado: “Es una de las herramientas más comunes que incluyen soporte para la detección de errores de programación y otras utilidades”

J

- **JDT** (*Java Development Toolkit*) o “Kit de Herramientas para Desarrolladores Java”
- **JDK** (*Java Development Kit*) o Kit de Desarrollo Java: “Es un software que provee herramientas de desarrollo para la creación de programas en Java”

- **JVM** (*Java Virtual Machine*) o Máquina Virtual de Java: “Es capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial , el cual es generado por el compilador del lenguaje Java”

K

- **Kernel de Linux:** “El kernel ó núcleo de linux es el corazón de este sistema operativo. Es el encargado de que el software y el hardware de un ordenador puedan trabajar juntos.”

L

- **Licencia BSD** (*Berkeley Software Distribution*) o “Esta licencia tiene menos restricciones en comparación con la GPL, permite el uso del código fuente en software no libre.”
- **Licencia GPL** (*General Public License*) o Licencia Publica General: “está orientada principalmente a proteger la libre distribución, modificación y uso de software

M

- **Máquina Virtual:** “Es un software que emula a una computadora y puede ejecutar programas como si fuese una computadora real”

P

- **Plugin:** “Es un complemento o aplicación adicional que es ejecutada por la aplicación principal”

S

- **SDK** (*Software Development kit*) o kit de desarrollo de software: “Es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto”

- **SGML** (*Standard Generalized Markup Language*) o Estándar de Lenguaje de Marcado Generalizado: “Consiste en un sistema para la organización y etiquetado de documentos”
- **SKINS**: “Elemento que permite cambiar la apariencia externa de la interfaz de una aplicación sin afectar su funcionalidad. Los skins suelen cambiar colores, formas y agregar efectos e imágenes a los programas.”
- **SQLite**: “Es un gestor de base de datos el cual fue creado especialmente para trabajar con celulares bajo Android”

U

- **UMTS** (Universal Mobile Telephone Service) o Sistema Universal de Telecomunicaciones Móviles: “Es una de las tecnologías usadas por los móviles de tercera generación”

X

- **XML** (*Extensible Markup Language*) o Lenguaje de Marcas Extensible: “XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.”

3.10.Referencias Bibliográficas

Bibliografía Consultada

- ✓ BERNAL, Cesar Augusto: *Metodología de la Investigación*. México: El Ateneo. 2006, Pág. 56.
- ✓ GARCÍA, Rosana: *Dispositivos Móviles*. 1ra edición. Ecuador: Humanistas. 2008, Pág.01.
- ✓ Microsoft ® Encarta ® 2007. © 1993--2006 Microsoft Corporation.
- ✓ RAMIRES, Enrique: *Desarrollo de Aplicaciones para Dispositivos con Sistema Operativo Android*. 1era Edición, Chile: Alfaomega. 2009, Pág. 10
- ✓ STALLMAN, Richard: *Introducción al Software Libre*. 1ra Edición. Quito-Ecuador:Carrillo.2011, Pág. 5, 9, 25
- ✓ UTC: *Reglamento de graduación a nivel de pregrado.Latacunga,2012*
- ✓ ZEA, Leiva: *Nociones de Metodología de Investigación Científica*. Quinta edición, 2001 Pág. 102.

Bibliografía Citada

- ✓ AGUSTIN, Fraude: *Programación con Java Script*. 2daEdición. Madrid:Macdril. 2008, Pág. 100, 112.
- ✓ ANDER-REG, Ezequiel: *Técnicas de Investigación-Social*. 24ava Edición. Buenos Aires: Humanistas.1989, Pág. 21, 22, 35,36.
- ✓ ARANAZ, Jaime: *Programación para Dispositivos Móviles*. España: 2005, Pág.98
- ✓ SLIDESHARE.NET [en línea]. 13 de Enero del 2012, <><http://www.slideshare.net/soreygarcia/consideraciones-basicas-para-el-desarrollo-de-aplicaciones-mviles>. [13 de Enero del 2012]
- ✓ AGILEDICTEMASANV [en línea]. Metodologías Agiles. 12 de Diciembre del 2011, <>http://www.Adamwesterski.com/wpcontent/files/docsCursos/Agiledoc_TemasAnv.pdf. [12 de Diciembre del 2011]

Bibliografía Virtual

- ✓ ABCPEDIA.COM [en línea]. Ciencia y Tecnología. 18 de julio del 2011, <><http://www.abcpedia.com/cienciaytecnologia/telefonoscelulares.html>. [Consulta: 18 de julio del 2011].
- ✓ WIKIPEDIA.ORG [en línea]. Tecnología XML. 21 de julio del 2011, <>http://es.Wikipedia.org/wiki/Extensible_Markup_Language. [Consulta: 21 de julio del 2011]
- ✓ SERVIDORES.COM [en línea]. Base de Datos Sqlite. 20 de julio del 2011, <><http://www.garabatoslinux.info/2011/01/sqlite-una-base-de-datos-sin-servidor.html>. [Consulta: 20 de julio del 2011]
- ✓ EVELIUX.COM [en línea]. Evolución de la Telefonía Móvil. 22 de Noviembre del 2011, <> <http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-móvil.php>. [Consulta: 22 de Noviembre del 2011].
- ✓ HISPALINUX.ES [en línea]. Software Libre. 26 de Noviembre del 2011, <><http://www.hispalinux.es/SoftwareLibre>. [26 de Noviembre del 2011]
- ✓ BLOGS.RPP.COM.PE [en línea]. Sistemas Operativos para Celulares. 28 de Noviembre del 2011, <<http://blogs.rpp.com.pe/technovida/2011/03/15/%C2%BFde-que-se-trata-todo-esto-de-android-en-mi-telefono/>>. [28 de Noviembre del 2011]
- ✓ JAVA.COM [en línea]. Lenguaje Java. 05 de Diciembre del 2011, <>http://java.ciberaula.com/articulo/que_es_java/. [05 de Diciembre del 2011]
- ✓ ANDROIDEITY.COM [en línea]. Máquina Virtual Dalvik. 08 de Diciembre del 2011, <><http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>. [08 de Diciembre del 2011]
- ✓ ECURED [en línea]. Definición de Eclipse. 12 de Diciembre del 2011, <><http://www.cursosporinternet.info/index.php/the-news/43-programacion/167-los-ides-de-programacion.html>. [12 de Diciembre del 2011]

3.11. Anexos

ANEXOS

UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

ENCUESTA DIRIGIDA A: Estudiantes y Docentes de la Carrera de Ingeniería en Informática y Sistemas Computacionales.

OBJETIVO: Realizar un análisis y estudio de las herramientas libres para el desarrollo de aplicaciones móviles en teléfonos celulares con sistema operativo Android.

INSTRUCCIONES: Marque con una **X** la respuesta correcta.

1. ¿Cree usted que el desarrollo de aplicaciones móviles es lo más novedoso en la actualidad?

SI

NO

2. ¿Ha incursionado alguna vez en el desarrollo de aplicaciones móviles?

SI

NO

3. ¿Ha descargado e instalado aplicaciones para su teléfono celular que tenga sistema operativo Android?

SI

NO

4. ¿Qué tipos de licencias considera usted utilizar para desarrollar aplicaciones para dispositivos móviles?

Software Libre

Software Propietario

5. ¿Para qué fin ha utilizado las aplicaciones de su celular?

Oficina Móvil
Marketing

Entretenimiento
Inventarios

6. ¿Cuál considera usted que es un sistema operativo para dispositivos móviles?

Windows 7	<input type="checkbox"/>	Windows XP	<input type="checkbox"/>
Linux Centos	<input type="checkbox"/>	As-400	<input type="checkbox"/>
Novell	<input type="checkbox"/>	Android	<input type="checkbox"/>

7. ¿Ha desarrollado aplicaciones para dispositivos móviles. Especifique en que lenguaje de programación?

C#	<input type="checkbox"/>	Java	<input type="checkbox"/>
Perl	<input type="checkbox"/>	Php	<input type="checkbox"/>
Visual Basic	<input type="checkbox"/>	No Conoce	<input type="checkbox"/>

8. ¿Ayudaría en su aprendizaje conocer el proceso de cómo se desarrollan aplicaciones para dispositivos móviles (celular)?

SI NO

9. ¿Le gustaría conocer nuevas herramientas para el desarrollo de aplicaciones móviles sobre un sistema operativo Android?

SI NO

10. ¿Le gustaría crear o desarrollar sus propias aplicaciones (software) para su celular?

SI NO

GRACIAS POR SU COLABORACIÓN...

ANEXO N° 3.2. Celulares con Android

MARCA DE CELULAR	SISTEMA OPERATIVO	OBSERVACIONES DE SELECCIÓN
LG HTC HUAWEI SAMSUNG MOTOROLA SONY ERICSON SMARTPHONE DELL SMARTPNONE ACER	Android	Como se puede observar el sistema operativo Android se encuentra en la mayor parte de celulares de las marcas mas conocidas, por esta razón fundamento que el tema de tesis se desarrolle en celulares con sistema operativo Android
IPHONE	iOS	Sistema Cerrado
LG NOKIA SAMSUNG MOTOROLA SONY ERICSON	Symbian	Sistema que no abarca muchas mas marcas de telefonos por su dificil instalacion de aplicaciones en los dispositivos.
BLACKBERRY	BLACKBERRY OS	Sistema Cerrado
LG HTC SAMSUNG	Windows Phone	Nuevo Sistema aun no se encuentra muy difundido en el mercado

ELABORADO POR: La Autora

ANEXO N° 3.3. Lenguajes de Programación

LENGUAJE DE PROGRAMACIÓN	CARACTERISTICAS	OBSERVACIONES DE SELECCIÓN
BASIC4 ANDROID	<ul style="list-style-type: none"> ✓ Para programadores en visual basic ✓ Rival de Java ✓ No es el mismo lenguaje de Microsoft 	No se encuentra mucho soporte técnico para este lenguaje. No es el mismo lenguaje de Microsoft, pero su sintaxis es la misma, lo cual tiene sus mismas ventajas como algunos de sus inconvenientes.
MONO PARA ANDROID	<ul style="list-style-type: none"> ✓ Empresa creadora Xamarin ✓ Lenguajes de programación de Microsoft como C# ✓ Costo de obtención 	Desventaja por costo de obtención
APP INVENTOR	<ul style="list-style-type: none"> ✓ Programación Grafica Drag and Drop ✓ Muy útil para desarrollo de aplicaciones estáticas ✓ Sin costo de obtención 	Desventaja por costo de obtención
JAVA	<ul style="list-style-type: none"> ✓ Nativo de Android ✓ Útil para desarrollo de aplicaciones dinámicas ✓ Soporte técnico abundante en la nube ✓ Programación orientada a objetos muy utilizado a nivel mundial ✓ Google utiliza java por defecto ✓ Sin costo de obtención 	Se selecciono este lenguaje de programación debido a que es nativo de Android y no tiene costo, y porque se utiliza la programación orientada a objetos que es utilizada en la mayor parte de desarrollo de aplicaciones de diferente tipo, y además que ayuda en el desarrollo de aplicaciones dinámicas, es decir acceso con base de datos.
LIVE CODE	<ul style="list-style-type: none"> ✓ Lenguaje para trabajar en varias plataformas como: Windows Phone, iOS, Android, Linux ✓ Trabaja en base a eventos ✓ Costo de obtención 	Desventaja por costo de obtención
IN DESIGN CS6	<ul style="list-style-type: none"> ✓ Desarrollo para aplicaciones estáticas ✓ Costo de obtención ✓ Multiplataforma 	Desventaja por costo de obtención

ELABORADO POR: La Autora

ANEXO N° 3.4. IDES de Programación

IDES DE PROGRAMACIÓN	CARACTERISTICAS	OBSERVACIONES DE SELECCIÓN
JBuilder	✓ Software Comercial	Desventaja por costo de obtención
JCreator	✓ Software Comercial	Desventaja por costo de obtención
Eclipse	<ul style="list-style-type: none"> ✓ Software libre ✓ Es uno de los entornos java mas utilizados a nivel profesional ✓ Facilita bastante el desarrollo de aplicaciones java por la funcionalidad que permite adicionar a través de plugins, es decir hay mas plugins para eclipse. 	Se ha seleccionado este ide debido a que no involucra en un 100% el diseño de las aplicaciones sino la funcionalidad de las mismas, ya que existe plugins para extender dicha funcionalidad de acuerdo al tipo de aplicación a desarrollar.
Netbeans	<ul style="list-style-type: none"> ✓ Software libre ✓ Es uno de los entornos java mas utilizados a nivel profesional ✓ Facilita bastante el diseño gráfico asociado a aplicaciones Java. 	Es mas utilizado porque facilita el diseño grafico asociado a aplicaciones java, y un porcentaje no tan menor a la funcionalidad
JDeveloper	<ul style="list-style-type: none"> ✓ No es rico en funcionalidades ✓ Es de Oracle 	Desventaja por costo de obtención
BlueJ	<ul style="list-style-type: none"> ✓ Es un entorno de desarrollo dirigido al aprendizaje de Java (entorno académico), por lo que tiene su limitante en funcionalidad y diseño grafico 	Limitante porque solo es para aprendizaje académico y no profesional

ELABORADO POR: La Autora

ANEXO N° 3.5. Bases de Datos Embebidas

BASES DE DATOS EMBEBIDAS	CARACTERISTICAS	OBSERVACIONES DE SELECCIÓN
ScimoreDB	<ul style="list-style-type: none"> ✓ Admite hasta 512 ordenadores en un clúster de la base de datos única y ofrece el valor en términos de funcionalidad, velocidad y costo. 	Desventaja por costo de obtención
SQL Server Compact	<ul style="list-style-type: none"> ✓ SQL Server Compact únicamente soporta tipos de datos de cadena compatibles ✓ No soporta perfiles de usuario ✓ Tamaño limitado en el archivo de base de datos ✓ Compatible solo con aplicaciones propietarias 	Desventaja por la compatibilidad solo con herramientas propietarias
Embended Firebird Server	<ul style="list-style-type: none"> ✓ Base de datos relacional ✓ Lenguaje de consulta SQL ✓ Código abierto ✓ Escrito en C++ ✓ Específicamente excelente para la creación de catálogos en CD-Rom ✓ Base de datos de solo lectura. 	Su tratamiento en la mayor parte esta en la creación de base de datos solo de lectura.
SQLite	<ul style="list-style-type: none"> ✓ Escrita en C ✓ Soporte de tablas e índices en un único archive ✓ Soporte transaccional ✓ Rapidez y escaso tamaño 25 mil líneas de código y en las consultas bloque a los demás usuarios ✓ Portabilidad ✓ Soporta SQL ✓ Base de datos relacional ✓ Soporta perfil de usuarios 	Se selecciono esta base de datos debido a que es nativo del sdk de Android, y además tiene incorporado librerías para su fácil manejo transaccional, y soporta todo lo habitual de los grandes motores de bases de datos.
Oracle Berkeley	<ul style="list-style-type: none"> ✓ Escrita en C ✓ Necesita licencia commercial ✓ Necesita de una capa XQuery para escribir documentos XML ✓ No soporta SQL o esquemas ✓ Ocupa mayor espacio el doble SQLite 	Desventaja por costo de obtención

Gadfly	<ul style="list-style-type: none"> ✓ Implementada en Python ✓ Requiere instalación ✓ Necesita licencia ✓ Soporta SQL ✓ Base datos relacional 	Desventaja por costo de obtención
Timesten	<ul style="list-style-type: none"> ✓ Base de datos relacional ✓ Se utiliza como registro o cache de bdd Oracle ✓ Las consultas se ejecutan mediante SQL92 o PL/SQL ✓ Necesita de muchas APIS se vuelve lento ✓ Necesita de licencia ✓ Recomendado para dispositivos de conectividad como CISco 	Desventaja por costo de obtención
<u>Metakit</u>	<ul style="list-style-type: none"> ✓ Es portable utiliza los datos sobre cualquier plataforma ✓ Especificamente utilizado en OS X de Mac ✓ Archivo sin formato ✓ Relacional ✓ Orientada a Objetos ✓ APIS Mk4py Python, Mk4tcl apis complejas para la conexión 	Complejidad en la conexión con las APIS Mk4py Python, Mk4tcl

ELABORADO POR: La Autora

ANEXO N° 3.6. Dispositivos Virtuales para Android

MAQUINA VIRTUAL PARA ANDROID	CARACTERISTICAS	OBSERVACIONES DE SELECCIÓN
DALVIK	<p>Dalvik es distribuida como software libre (bajo licencia Apache 2.0) fue diseñada por Dan Bornstein junto con otros diseñadores de Google específicamente para el sistema operativo de móviles/tabletas Android, está optimizada para tener bajos requerimientos de memoria y permitir múltiples instancias de la VM funcionando de manera simultánea, dejando para el sistema operativo subyacente tareas como la gestión de la memoria y la gestión de procesos e hilos.</p> <p>Dalvik utiliza un bytecode muy similiar pero no igual que el Java Byte Code por lo que en la práctica no son compatibles, por lo que es necesario que las clases .class sean transformadas por una utilidad llamada dx (incluida en el Android SDK) que las transforma en el Bytecode de Dalvik en clases .dex, las cuales finalmente son comprimidas en ficheros .apk que son los que conforman los paquetes de Android.</p>	<p>Es la maquina virtual que viene ya incluida en el paquete del sdk de Android, y es la única Maquina Virtual que se encontró para su aplicación.</p>

ELABORADO POR: La Autora