



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**ANÁLISIS COMPARATIVO ENTRE LAS PLATAFORMAS DE DESARROLLO
DE APLICACIONES MÓVILES PARA LOS SISTEMAS OPERATIVOS
ANDROID Y IOS.**

TESIS DE GRADO

Previa la obtención del título de

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

KARLA JOHANNA CARVAJAL VALDIVIESO

CHRISTIAN FERNANDO SASIG SUNTASIG

RIOBAMBA – ECUADOR

2012

DEDICATORIA

Mi tesis la dedico con todo mi amor y cariño:

A Dios por brindarme salud y vida para alcanzar todos los objetivos que me he propuesto en esta etapa de mi vida.

A mis padres y hermanos por ser un apoyo incondicional y por tenerme paciencia en los buenos y malos momentos académicos y en mi vida diaria.

A mis amigos por estar a mi lado y por su apoyo.

A mis maestros por ser una guía y por transmitirme sus conocimientos que son de mucho valor para continuar ahora en el camino profesional.

Gracias a todos...

Karla Johanna Carvajal Valdívieso.

DEDICATORIA

Agradezco a Dios por permitirme llegar hasta donde estoy y por bendecirme con el don de la vida y con una familia tan maravillosa.

Dedico mi tesis a mis padres Rosa y Segundo por su infinito e incondicional amor. Gracias Mami por tu ayuda, tu paciencia, por tu apoyo y tu amor. Papi gracias por tu entrega y dedicación para brindarme siempre lo mejor. Gracias a los dos por inculcarme el valor de la vida, de la verdad, el respeto y la responsabilidad, la fortaleza para siempre luchar por mis sueños. Este logro es gracias a Ustedes.

A mis hermanos William, Washington, por su motivación, paciencia y apoyo incondicional; y a mis amigos por su amistad sincera.

A todos mis profesores que colaboraron en mi formación académica. Sus lecciones permanecerán siempre conmigo.

Christian Fernando Sasig Suntasig.

NOMBRE

FIRMA

FECHA

Ing. Iván Menes

**DECANO DE LA FACULTAD
DE INFORMÁTICA Y
ELECTRÓNICA**

.....

.....

Ing. Raúl Rosero

**DIRECTOR DE LA
ESCUELA DE INGENIERÍA
EN SISTEMAS.**

.....

.....

Ing. Danilo Pástor

DIRECTOR DE TESIS

.....

.....

Ing. Gloria Arcos

MIEMBRO DEL TRIBUNAL

.....

.....

Lcdo. Carlos Rodríguez

**DIRECTOR DPTO.
DOCUMENTACIÓN**

.....

.....

NOTA DE LA TESIS

.....

Nosotros, **Karla Johanna Carvajal Valdivieso y Christian Fernando Sasig Suntasig** somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

.....

Karla Johanna Carvajal Valdivieso

.....

Christian Fernando Sasig Suntasig

ÍNDICE DE ABREVIATURAS

ADT:	Android Development Tools
APK:	Android Package (Paquete de Android)
BSD:	Berkeley Software Distribution
DDMS:	Dalvik Debug Monitor Server
GPLV3:	Licencia Pública General de GNU
GUI:	Interfaz Gráfica de Usuario
HTC:	High Tech Computer
IDE:	Entorno de Desarrollo Integrado
JSON:	JavaScript Object Notation
NDK:	Kit de Desarrollo Nativo
NFC:	Near Field Communication
OS:	Sistema Operativo
RIM:	Research In Motion
SDK:	Kit de Desarrollo de Software
SMS:	Short Message Service
SOAP:	Simple Object Access Protocol
VGA:	Video Graphics Array
XAML:	Lenguaje Extensible de Formato para Aplicaciones
XML:	Lenguaje de Marcado Extensible
XP:	Extreme Programming

INDICE

CAPITULO I	20
MARCO REFERENCIAL	20
1.1 TÍTULO DEL PROYECTO DE TESIS:.....	20
1.2 LUGAR DE REALIZACIÓN	20
1.3 ANTECEDENTES	20
1.4 JUSTIFICACIÓN	23
1.4.1 JUSTIFICACIÓN TEÓRICA.....	23
1.4.2 JUSTIFICACIÓN APLICATIVA	24
1.5 OBJETIVOS	26
1.5.1 OBJETIVO GENERAL.....	26
1.5.2 OBJETIVOS ESPECÍFICOS.....	26
1.6 HIPÓTESIS	26
CAPITULO II	27
INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES	27
INTELIGENTES	27
2.1 INTRODUCCION.....	27
2.2 HISTORIA DE LOS DISPOSITIVOS MÓVILES INTELIGENRTES.....	30
2.2.1 PRIMEROS TELÉFONOS INTELIGENTES.....	30
2.3 MERCADO DE LOS DISPOSITIVOS MÓVILES INTELIGENTES	35
2.4 SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES INTELIGENTES...38	
2.4.1 BLACKBERRY OS	39
2.4.1.1 DESARROLLANDO PARA RIM.....	40
2.4.2 SYMBIAN	40
2.4.2.1 DESARROLLANDO EN SYMBIAN OS	41
2.4.3 WINDOWS PHONE 7	43
2.4.3.1 DESARROLLANDO EN WINDOWS PHONE 7.....	44
2.4.4 ANDROID.....	45
2.4.4.1 DESARROLLANDO PARA ANDROID	47
2.4.5 IOS.....	47
2.4.5.1 DESARROLLANDO PARA IOS	48
2.4.6 MEEGO	48
CAPITULO III	50
ANÁLISIS COMPARATIVO	50
3.1 INTRODUCCIÓN.....	50

3.2 DETERMINAR LAS PLATAFORMAS DE DESARROLLO PARA DISPOSITIVOS MÓVILES INTELIGENTES A COMPARAR	50
3.3 ANÁLISIS DE LAS PLATAFORMAS SELECCIONADAS	53
3.3.1 ANDROID	53
3.3.1.1 CARACTERÍSTICAS	55
3.3.1.2 RESTRICCIONES	56
3.3.1.3 VERSIONES DE LA PLATAFORMA.....	56
3.3.1.4 ARQUITECTURA.....	58
3.3.1.5 SDK (KIT DE DESARROLLO DE SOFTWARE)	67
3.3.1.6 SISTEMAS OPERATIVOS.....	70
3.3.1.7 LENGUAJES DE PROGRAMACIÓN	70
3.3.1.8 ENTORNOS DE PROGRAMACIÓN	71
3.3.1.9 DISTRIBUCIÓN DE APLICACIONES	72
3.3.1.10 TIENDA DE APLICACIONES	72
3.3.1.11 INSTALACIÓN DE ECLIPSE CON EL SDK DE ANDROID	73
3.3.2 IOS	75
3.3.2.1 CARACTERÍSTICAS	76
3.3.2.2 RESTRICCIONES	79
3.3.2.3 VERSIONES DE IOS.....	80
3.3.2.4 ARQUITECTURA DE IOS.....	80
3.3.2.5 SDK DE IOS	83
3.3.2.6 SISTEMAS OPERATIVOS.....	85
3.3.2.7 LENGUAJES DE PROGRAMACIÓN	85
3.3.2.8 ENTORNO DE PROGRAMACIÓN	86
3.3.2.9 DISTRIBUCIÓN DE APLICACIONES	87
3.3.2.10 TIENDA DE APLICACIONES	89
3.3.2.11 INSTALACIÓN DE XCODE EN MAC OS X SNOW LEOPARD	90
3.4 DETERMINACIÓN DE LOS PARÁMETROS DE COMPARACIÓN	92
3.5 AMBIENTE DE PRUEBAS	97
3.6 DEMOSTRACIÓN DE PARÁMETROS DE COMPARACIÓN	97
3.6.1 DESCRIPCIÓN DE LOS MÓDULOS DE PRUEBA PRÁCTICOS	97
3.6.2 DESARROLLO DE LOS MÓDULOS DE PRUEBA.....	99
3.6.2.1 MÓDULO TEÓRICO DE LOS PARÁMETROS LENGUAJES DE PROGRAMACIÓN Y SISTEMAS OPERATIVOS	99
3.6.2.2 MÓDULOS DE PRUEBA PRÁCTICOS DEL PARÁMETRO INTERFAZ GRÁFICA DE USUARIO Y PERSISTENCIA DE DATOS	100
3.6.2.2.1 MÓDULO 1 EN ANDROID.....	100

3.6.2.2.2 MÓDULO 1 EN IOS	112
3.6.2.3 MÓDULOS DE PRUEBA DEL PARÁMETRO ACCESO A SERVICIOS WEB 126	
3.6.2.3.1 MÓDULO 2 EN ANDROID	126
3.6.2.3.2 MÓDULO 2 EN IOS	131
3.6.2.4 MÓDULOS DE PRUEBA DE LOS PARÁMETROS ENTORNO DE DESARROLLO, COMPATIBILIDAD ENTRE VERSIONES Y FACILIDADES DE DESPLIEGUE.	138
3.6.2.4.1 MÓDULO 3 EN ANDROID	138
3.6.2.4.2 MÓDULO 3 EN IOS	152
3.7 ANÁLISIS COMPARATIVO	166
3.7.1 LENGUAJE DE PROGRAMACIÓN	169
3.7.2 SISTEMAS OPERATIVOS	171
3.7.3 INTERFAZ GRÁFICA DE USUARIO	172
3.7.4 PERSISTENCIA DE DATOS	174
3.7.5 ACCESO A SERVICIOS WEB.....	176
3.7.6 ENTORNO DE DESARROLLO INTEGRADO.....	178
3.7.7 COMPATIBILIDAD ENTRE VERSIONES	180
3.7.8 FACILIDADES DE DESPLIEGUE.....	182
3.7.9 PUNTAJES ALCANZADOS.....	184
3.8 DIAGRAMA GENERAL DE RESULTADOS.....	186
3.8.1 INTERPRETACIÓN.....	187
3.8.2 RESULTADO DEL ANÁLISIS.....	188
3.9 COMPROBACIÓN DE HIPÓTESIS	189
3.9.1 CONCLUSIÓN.....	190
CAPITULO IV	191
DESARROLLO DE LA APLICACIÓN PARA ATENCIÓN DE CLIENTES EN LA RECOLECCIÓN DE PEDIDOS EN BONNY RESTAURANT	191
4.1. PARTE TEÓRICA.....	191
4.1.1. INTRODUCCIÓN.....	191
4.1.2. DESARROLLO	192
4.2. PARTE PRÁCTICA.....	198
4.2.1. HERRAMIENTAS UTILIZADAS.....	198
4.2.2. INTRODUCCIÓN.....	199
4.2.3. DEFINICIÓN DEL ÁMBITO	199
4.2.4. EQUIPO DE TRABAJO	200
4.2.5. SERVICIOS	200

4.2.6. FUNCIONES	200
4.2.7. DESCRIPCIÓN DEL SISTEMA ACTUAL	201
4.2.8. IDENTIFICACIÓN DEL PROBLEMA	201
4.2.9. FASES.....	201
4.2.9.1. PLANEACIÓN.....	201
4.2.9.1.1 HISTORIAS DE USUARIO	201
4.2.9.1.2 PLAN DE PUBLICACIONES.....	202
4.2.9.1.3 VELOCIDAD DEL PROYECTO	202
4.2.9.1.4 ITERACIÓN	203
4.2.9.1.5 PROGRAMACIÓN EN PAREJAS	204
4.2.9.2. DISEÑO.....	205
4.2.9.2.1 DISEÑOS SIMPLES	205
4.2.9.2.2 RIESGOS.....	214
4.2.9.3. CODIFICACIÓN.....	219
4.2.9.4. PRUEBAS	219
CONCLUSIONES	221
RECOMENDACIONES	223
RESUMEN	224
ABSTRACT.....	225
ANEXOS.....	227
GLOSARIO	247
BIBLIOGRAFIA	249

INDICE DE FIGURAS

FIGURA I - 1. ARQUITECTURA DE LA APLICACIÓN	25
FIGURA II - 1. HTC DESIRE, EL SMARTPHONE MÁS POPULAR DE LOS TELÉFONOS MÓVILES ANDROID.....	28
FIGURA II - 2. IPHONE, EL SMARTPHONE DE APPLE.....	29
FIGURA II - 3. IBM SIMON (1992).....	30
FIGURA II - 4. NOKIA 9000 (1996).....	31
FIGURA II - 5. NOKIA 9210 CON PANTALLA A COLOR (2000).....	31
FIGURA II - 6. GS88 DE ERICSSON (1997).....	32
FIGURA II - 7. BLACKBERRY QUARK CON PANTALLA MONOCROMO.(2000)	32
FIGURA II - 8. BLACKBERRY BOLD 9900, CON TECLADO Y PANTALLA TÁCTIL (2011)	33
FIGURA II - 9. LANZAMIENTO DEL PRIMER IPHONE EN EEUU (2007)	34
FIGURA II - 10. EL PRIMER SMARTPHONE CON ANDROID: HTC DREAM (OCTUBRE 2008)	34
FIGURA II - 11. ESTUDIO DE VISION MOBILE	36
FIGURA III - 1 SISTEMAS OPERATIVOS MÁS UTILIZADOS EN DISPOSITIVOS MÓVILES INTELIGENTES	51
FIGURA III - 2 CUOTA DE MERCADO DE SISTEMAS OPERATIVOS EN ESTADOS UNIDOS	52
FIGURA III - 3 ARQUITECTURA DE ANDROID	59
FIGURA III - 4 CICLO DE VIDA DE UNA APLICACIÓN EN ANDROID.....	66
FIGURA III - 5 ARQUITECTURA DE MAC OSX Y IPHONE OS.....	81
FIGURA III - 6 ESQUEMA DEL SDK DE IOS.....	84
FIGURA III - 7 XCODE.....	90
FIGURA III - 8 DESCARGA DEL XCODE	90
FIGURA III - 9 ACUERDO DE LICENCIA DE XCODE.....	91
FIGURA III - 10 INSTALACIÓN COMPLETADA	91
FIGURA III - 11 INSTALACIÓN COMPLETADA	97
FIGURA III - 12 CREACIÓN DE UN NUEVO PROYECTO.....	101
FIGURA III - 13 SELECCIÓN DE LA VERSIÓN DE ANDROID.....	102
FIGURA III - 14 DIRECTORIOS DE CARPETAS PARA LAS INTERFACES EN ANDROID ..	102
FIGURA III - 15 PALETA DE ELEMENTOS DE LA INTERFAZ GRÁFICA.....	103
FIGURA III - 16 CÓDIGO XML DE LA INTERFAZ DE USUARIO.....	104
FIGURA III - 17 CÓDIGO JAVA DE LA INTERFAZ DE USUARIO	104
FIGURA III - 18 LLAMADA A LA VISTA MAIN	104
FIGURA III - 19 VINCULACIÓN DEL CÓDIGO JAVA CON LOS ELEMENTOS DE LA VISTA	105
FIGURA III - 20 PROPIEDAD ONCLICK EN EL CÓDIGO XML.....	105
FIGURA III - 21 PROPIEDAD ONCLICK EN EL CÓDIGO JAVA.....	105

FIGURA III - 22 CREACIÓN DE UNA BASE DE DATOS CON SQLITE MANAGER	106
FIGURA III - 23 CREACIÓN DE UNA BASE DE DATOS CON CÓDIGO JAVA.....	107
FIGURA III - 24 DIRECTORIO DE LA BASE DE DATOS EN EL DISPOSITIVO.....	107
FIGURA III - 25 REGISTROS DE LA TABLA EN EL SQLITE MANAGER.....	108
FIGURA III - 26 CÓDIGO JAVA PARA LA INSERCIÓN DE REGISTROS	108
FIGURA III - 27 PANTALLA DE INGRESO DE DATOS.....	108
FIGURA III - 28 LISTADO DE DATOS DEL CONTACTO	109
FIGURA III - 29 CÓDIGO JAVA PARA LA ACTUALIZACIÓN DE REGISTROS	109
FIGURA III - 30 PANTALLA DE ACTUALIZACIÓN DE DATOS.....	109
FIGURA III - 31 LISTADO DE DATOS DEL CONTACTO	110
FIGURA III - 32 CÓDIGO JAVA PARA LA ELIMINACIÓN DE REGISTROS.....	110
FIGURA III - 33 BÚSQUEDA DEL REGISTRO A ELIMINAR.....	110
FIGURA III - 34 LISTADO DE DATOS DEL CONTACTO	111
FIGURA III - 35 CÓDIGO JAVA PARA RECUPERAR DATOS DE LOS REGISTROS	111
FIGURA III - 36 LISTADO DE DATOS DEL CONTACTO	111
FIGURA III - 37 PANTALLA DE CREACIÓN DE UN NUEVO PROYECTO IOS CON XCODE	112
FIGURA III - 38 ESTABLECER EL NOMBRE DEL PROYECTO	113
FIGURA III - 39 INTERFAZ GRÁFICA DEL MÓDULO DE PERSISTENCIA	113
FIGURA III - 40 HERRAMIENTAS PARA LA INTERFAZ GRÁFICA.....	114
FIGURA III - 41 DEFINICIÓN DE LA INTERFAZ GRÁFICA MEDIANTE CÓDIGO OBJECTIVE-C.....	115
FIGURA III - 42 CREACIÓN DE LA CLASE MODULOPERSISTENCIAVIEWCONTROLLER.H	115
FIGURA III - 43 CREACIÓN DE LA CLASE MODULOPERSISTENCIAVIEWCONTROLLER.M	116
FIGURA III - 44 VINCULACIÓN DE LA INTERFAZ DE USUARIO CON LAS PROPIEDADES DE LA CLASE	117
FIGURA III - 45 VINCULACIÓN DE MÉTODOS DE LA CLASE CON LOS EVENTOS	118
FIGURA III - 46 IMPLEMENTACIÓN DE EVENTOS MEDIANTE CÓDIGO OBJECTIVE-C ...	118
FIGURA III - 47 CREACIÓN DE BASE DE DATOS CON SQLITE MANAGER.....	119
FIGURA III - 48 CREACIÓN DE LAS TABLAS DE LA BASE DE DATOS	119
FIGURA III - 49 CREACIÓN DE LAS TABLAS DE LA BASE DE DATOS MEDIANTE CÓDIGO OBJECTIVE-C.....	120
FIGURA III - 50 ARCHIVO DE LA BASE DE DATOS CONTACTS.DB	121
FIGURA III - 51 TABLA DE LA BASE DE DATOS CONTACTS.....	121
FIGURA III - 52 CÓDIGO OBJECTIVE-C PARA LA INSERCIÓN DE REGISTROS	122
FIGURA III - 53 PANTALLA PARA EL INGRESO DE DATOS.....	122
FIGURA III - 54 LISTADO DE CONTACTOS	123
FIGURA III - 55 CÓDIGO OBJECTIVE-C PARA LA ACTUALIZACIÓN DE REGISTROS	123

FIGURA III - 56 PANTALLA DE EDICIÓN DE CONTACTOS	124
FIGURA III - 57 CÓDIGO OBJECTIVE-C PARA LA ELIMINACIÓN DE REGISTROS.....	124
FIGURA III - 58 BÚSQUEDA DEL REGISTRO A ELIMINAR.....	125
FIGURA III - 59 RESULTADO DE LA ELIMINACIÓN DE REGISTROS.....	125
FIGURA III - 60 CÓDIGO OBJECTIVE-C PARA LA RECUPERACIÓN DE REGISTROS.....	126
FIGURA III - 61 LISTADO DE CONTACTOS	126
FIGURA III - 62 CREACIÓN DEL MÓDULO SERVICIO WEB.....	127
FIGURA III - 63 CREACIÓN DEL SERVICIO WEB.....	128
FIGURA III - 64 CÓDIGO JAVA DEL CONSUMO DE UN SERVICIO WEB CON JSON	129
FIGURA III - 65 CÓDIGO JAVA DE LA EJECUCIÓN DEL MÉTODO	129
FIGURA III - 66 PERMISO DE INTERNET EN EL ARCHIVO ANDROIDMANIFEST.XML	129
FIGURA III - 67 RESULTADO DEL CONSUMO DE DATOS MEDIANTE JSON.....	130
FIGURA III - 68 SERVICIO WEB GETCATEGORIES.....	130
FIGURA III - 69 CÓDIGO JAVA DEL CONSUMO DE DATOS MEDIANTE SOAP.....	131
FIGURA III - 70 RESULTADO DEL CONSUMO DE DATOS MEDIANTE SOAP	131
FIGURA III - 71 CREACIÓN DE UN PROYECTO EN XCODE.....	132
FIGURA III - 72 VENTANA DE CREACIÓN DEL PROYECTO EN XCODE	132
FIGURA III - 73 CÓDIGO DEL SERVICIO WEB CON PHP.....	133
FIGURA III - 74 LIBRERÍAS DE JSON JSONKIT.H Y JSONKIT.M	134
FIGURA III - 75 CÓDIGO DEL CONSUMO DEL SERVICIO WEB EN IOS.....	135
FIGURA III - 76 RESULTADO DEL CONSUMO DEL SERVICIO WEB.....	135
FIGURA III - 77 SERVICIO WEB CON SOAP GET-CATEGORIES	136
FIGURA III - 78 CONSUMO DEL SERVICIO WEB CON SOAP	137
FIGURA III - 79 RESULTADO DEL CONSUMO DEL SERVICIO WEB CON SOAP.....	138
FIGURA III - 80 AYUDA INTELIGENTE EN ECLIPSE Y DOCUMENTACIÓN DE ANDROID	139
FIGURA III - 81 CONJUNTO DE CODE SNIPPETS INCORPORADOS CON EL SDK	140
FIGURA III - 82 EJEMPLO DE CÓDIGO IMPLEMENTADO MEDIANTE EL SDK	140
FIGURA III - 83 EJEMPLO DE CÓDIGO IMPLEMENTADO MEDIANTE ECLIPSE.....	141
FIGURA III - 84 VENTANA DE DEPURACIÓN DE CÓDIGO EN ECLIPSE	142
FIGURA III - 85 ELEMENTOS DE DISEÑO DE LA INTERFAZ GRÁFICA PARA ANDROID .	143
FIGURA III - 86 ELEMENTOS DESDE LA PALETA DE COMPONENTES HACIA EL PANEL	143
FIGURA III - 87 CONFIGURACIÓN DE LA VERSIÓN DEL SDK MEDIANTE XML	144
FIGURA III - 88 CONFIGURACIÓN DE LA VERSIÓN DEL SDK MEDIANTE INTERFAZ GRÁFICA	144
FIGURA III - 89 APLICACIÓN HONEYCOMBGALLERY EN SMARTPHONE.....	145
FIGURA III - 90 APLICACIÓN HONEYCOMBGALLERY EN TABLET	146
FIGURA III - 91 VENTANA PARA LA ADMINISTRACIÓN DE DISPOSITIVOS VIRTUALES .	147
FIGURA III - 92 VENTANA DE CREACIÓN DE UN NUEVO DISPOSITIVO VIRTUAL.....	148
FIGURA III - 93 HABILITAR EL MODO DE DEPURACIÓN EN EL DISPOSITIVO	149

FIGURA III - 94 LISTA DE DISPOSITIVOS FÍSICOS CONECTADOS.....	150
FIGURA III - 95 EXPORTACIÓN DEL .APK DESDE ECLIPSE	151
FIGURA III - 96 HABILITAR FUENTES DESCONOCIDAS PARA APLICACIONES QUE NO SEAN DE GOOGLE PLAY.....	152
FIGURA III - 97 DOCUMENTACIÓN DE UN TIPO DE DATO NSMUTABLEARRAY.....	153
FIGURA III - 98 DETALLE DE LA DOCUMENTACIÓN DE UN NSMUTABLEARRAY.....	154
FIGURA III - 99 BLOQUE DE CÓDIGO DE UN SWITCH	154
FIGURA III - 100 VENTANA DE DEPURACIÓN DE CÓDIGO EN XCODE IOS	155
FIGURA III - 101 DISEÑADOR DE INTERFAZ GRÁFICA	156
FIGURA III - 102 LISTADO DE ELEMENTOS DE INTERFAZ GRÁFICA.....	156
FIGURA III - 103 CONFIGURACIÓN DEL ATRIBUTO BASE SDK	157
FIGURA III - 104 CONFIGURACIÓN DEL ATRIBUTO IOS DEPLOYMENT TARGET	158
FIGURA III - 105 APLICACIÓN DESARROLLADA PARA IPHONE	159
FIGURA III - 106 APLICACIÓN DESARROLLADA EN IPHONE EJECUTADA EN EL IPAD..	160
FIGURA III - 107 EMULADOR DEL IPAD.....	161
FIGURA III - 108 EMULADOR DEL IPAD.....	162
FIGURA III - 109 EL NOMBRE DEL PROYECTO ES MODULOSERVICIOSWEB	163
FIGURA III - 110 VENTANA DEVELOPER PROFILE.....	163
FIGURA III - 111 VENTANA AUTOMATIC DEVICE PROVISIONING	164
FIGURA III - 112 VENTANA DE DISPOSITIVOS.....	164
FIGURA III - 113 VENTANA DE LOGIN DE LA CUENTA DE DESARROLLADOR	164
FIGURA III - 114 VENTANA DE RECONOCIMIENTO DEL DISPOSITIVO.....	165
FIGURA III - 115 VENTANA ESQUEMAS EN XCODE.....	165
FIGURA III - 116 RESULTADO DEL PARÁMETRO LENGUAJES DE PROGRAMACIÓN.....	170
FIGURA III - 117 RESULTADO DEL PARÁMETRO SISTEMAS OPERATIVOS	172
FIGURA III - 118 RESULTADO DEL PARÁMETRO INTERFAZ GRÁFICA DE USUARIO	174
FIGURA III - 119 RESULTADO DEL PARÁMETRO PERSISTENCIA DE DATOS	176
FIGURA III - 120 RESULTADO DEL PARÁMETRO ACCESO A SERVICIO WEB.....	178
FIGURA III - 121 RESULTADO DEL PARÁMETRO ENTORNO DE DESARROLLO INTEGRADO.....	180
FIGURA III - 122 RESULTADO DEL PARÁMETRO COMPATIBILIDAD ENTRE VERSIONES	182
FIGURA III - 123 RESULTADO DEL PARÁMETRO FACILIDADES DE DESPLIEGUE	184
FIGURA III - 124 GRÁFICO GENERAL DE RESULTADOS	186
FIGURA III - 125 GRÁFICO GENERAL DE RESULTADOS	187
FIGURA IV - 1 USUARIO METODOLOGÍA XP	193
FIGURA IV - 2 LISTADO DE PLATOS	207
FIGURA IV - 3 LISTADO DE ORDENES.....	208
FIGURA IV - 4 LISTADO DE USUARIOS.....	208

FIGURA IV - 5 NAVEGACIÓN POR LAS DIFERENTES CATEGORÍAS.....	209
FIGURA IV - 6 DETALLE DE UN PLATO	209
FIGURA IV - 7 DETALLE DE LA ORDEN.....	210
FIGURA IV - 8 UBICACIÓN DE MESAS	210
FIGURA IV - 9 DIAGRAMA DE COMPONENTES	211
FIGURA IV - 10 ARQUITECTURA FÍSICA DE LA APLICACIÓN.....	212
FIGURA IV - 11 DIAGRAMA DE BASE DE DATOS.....	213

ÍNDICE DE TABLAS

TABLA II - I NÚMERO DE APLICACIONES POR PLATAFORMA	37
TABLA II - II PORCENTAJE DE DESARROLLADORES POR PLATAFORMA.....	38
TABLA III - I TIPOS DE RECURSOS EN ANDROID.....	67
TABLA III - II PARÁMETROS DE COMPARACIÓN.....	92
TABLA III - III VARIABLES DEL PARÁMETRO DE COMPARACIÓN LENGUAJES DE PROGRAMACIÓN	93
TABLA III - IV VARIABLES DEL PARÁMETRO DE COMPARACIÓN SISTEMAS OPERATIVOS	94
TABLA III - V VARIABLES DEL PARÁMETRO DE COMPARACIÓN INTERFAZ GRÁFICA DE USUARIO	94
TABLA III - VI VARIABLES DEL PARÁMETRO DE COMPARACIÓN PERSISTENCIA DE DATOS	94
TABLA III - VII VARIABLES DEL PARÁMETRO DE COMPARACIÓN ACCESO A SERVICIOS WEB	95
TABLA III - VIII VARIABLES DEL PARÁMETRO DE COMPARACIÓN ENTORNO DE DESARROLLO INTEGRADO	95
TABLA III - IX VARIABLES DEL PARÁMETRO DE COMPARACIÓN COMPATIBILIDAD ENTRE VERSIONES	96
TABLA III - X VARIABLES DEL PARÁMETRO DE COMPARACIÓN FACILIDADES DE DESPLIEGUE	96
TABLA III - XI TABLA COMPARATIVA ENTRE JAVA Y OBJETIVE C.....	99
TABLA III - XII ESCALA DE VALORACIONES.....	166
TABLA III - XIII ESCALA DE VALORACIONES CUALITATIVA	166
TABLA III - XIV PESOS PARA LOS PARÁMETROS	167
TABLA III - XV PORCENTAJES DE LOS RESULTADOS	168
TABLA III - XVI CALIFICACIÓN DE LAS TECNOLOGÍAS	168
TABLA III - XVII ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO LENGUAJES DE PROGRAMACIÓN.....	169
TABLA III - XVIII CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO LENGUAJES DE PROGRAMACIÓN	169

TABLA III - XIX ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO SISTEMAS OPERATIVOS.....	171
TABLA III - XX CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO SISTEMAS OPERATIVOS	171
TABLA III - XXI ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO INTERFAZ GRÁFICA DE USUARIO	172
TABLA III - XXII CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO INTERFAZ GRÁFICA DE USUARIO	172
TABLA III - XXIII ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO PERSISTENCIA DE DATOS	174
TABLA III - XXIV CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO PERSISTENCIA DE DATOS	174
TABLA III - XXV ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO ACCESO A SERVICIOS WEB.....	176
TABLA III - XXVI CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO ACCESO A SERVICIOS WEB	176
TABLA III - XXVII ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO ENTORNO DE DESARROLLO INTEGRADO	178
TABLA III - XXVIII CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO ENTORNO DE DESARROLLO INTEGRADO	178
TABLA III - XXIX ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO COMPATIBILIDAD ENTRE VERSIONES.....	180
TABLA III - XXX CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO COMPATIBILIDAD ENTRE VERSIONES.....	180
TABLA III - XXXI ESCALA DE VALORACIONES CUALITATIVA PARA EL PARÁMETRO FACILIDADES DE DESPLIEGUE.....	182
TABLA III - XXXII CALIFICACIÓN DE LAS VARIABLES DEL PARÁMETRO FACILIDADES DE DESPLIEGUE	182
TABLA III - XXXIII TABLA GENERAL DE RESULTADOS.....	185
TABLA IV - I CASO DE USO AUTENTICAR	230
TABLA IV - II CASO DE USO CERRAR SESIÓN	231
TABLA IV - III CASO DE USO AGREGAR CATEGORÍA DE PLATOS	232
TABLA IV - IV CASO DE USO AGREGAR CATEGORÍAS DE INGREDIENTES.....	233
TABLA IV - V CASO DE USO AGREGAR INGREDIENTE	234
TABLA IV - VI CASO DE USO AGREGAR PLATO.....	235
TABLA IV - VII CASO DE USO AGREGAR MESA	236
TABLA IV - VIII CASO DE USO LISTAR ORDENES DEL DÍA	237
TABLA IV - IX CASO DE USO EJECUTAR ORDEN.....	238
TABLA IV - XI CASO DE USO GENERAR REPORTE	240
TABLA IV - XII CASO DE USO AUTENTICAR.....	241
TABLA IV - XIII CASO DE USO CERRAR SESIÓN.....	242
TABLA IV - XIV CASO DE USO SELECCIONAR LA MESA.....	243
TABLA IV - XV CASO DE USO AGREGAR ORDEN	244

TABLA IV - XVI COSTOS TANGIBLES DEL PROYECTO.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA IV - XVII HISTORIA DE USUARIO 1	228
TABLA IV - XVIII HISTORIA DE USUARIO 2	228
TABLA IV - XIX HISTORIA DE USUARIO 3	229
TABLA IV - XX HISTORIA DE USUARIO 4	229
TABLA IV - XXI HISTORIA DE USUARIO 5	229
TABLA IV - XXII IDENTIFICACIÓN DE RIESGOS	214
TABLA IV - XXIII CRITERIOS DE VALORACIÓN DE LA PROBABILIDAD DE RIESGOS	214
TABLA IV - XXIV CRITERIO DE VALORACIÓN DEL IMPACTO DE RIESGOS	214
TABLA IV - XXV DETERMINACIÓN DE EXPOSICIÓN AL RIESGO.....	215
TABLA IV - XXVI VALORACIÓN DE DETERMINACIÓN DE EXPOSICIÓN AL RIESGO	215
TABLA IV - XXVII IMPACTO ESPERADOS	215
TABLA IV - XXVIII PRIORIZACIÓN DE LOS RIESGOS	216
TABLA IV - XXIX HOJA DE GESTIÓN DE RIESGO 1	216
TABLA IV - XXX HOJA DE GESTIÓN DE RIESGO 2	218

INTRODUCCIÓN

En la actualidad, las tecnologías móviles, y en concreto el mercado de los dispositivos móviles inteligentes, se encuentran en su mayor momento de expansión. Estos nuevos dispositivos móviles traen nuevas formas de comunicación entre personas, así como nuevas formas de interacción entre usuario y máquina, gracias a la evolución del hardware de los dispositivos móviles, su rango de uso se ha ampliado a varios campos, como el multimedia, el uso de sensores, el consumo de servicios web y la navegación por *Internet*.

Hoy en día es amplia la utilización de dispositivos móviles inteligentes y su adecuado uso ha permitido optimizar la gestión de los tiempos de trabajo e incrementar la productividad, todo ello ayudado de una gran portabilidad, característica fundamental de los dispositivos móviles inteligentes, que ha permitido que ellos sean parte de nuestra vida diaria

En concreto, dentro de este nuevo campo de desarrollo de aplicaciones, y la diversidad de sistemas operativos y plataformas de desarrollo existentes permiten mantener un alto grado de competencia en el mercado, lo cual es bueno para los consumidores, sin embargo es de gran impacto para las personas y empresas que dedican sus esfuerzos a la implementación de aplicaciones y servicios para dispositivos móviles. Este proyecto analiza la problemática del desarrollo de este tipo de aplicaciones para obtener una máxima productividad en la implementación, adaptándolo cada vez a las necesidades cambiantes de los clientes.

En el primer capítulo se detalla la información general del presente proyecto de tesis, en el segundo capítulo se encuentra el marco teórico necesario para el estudio de las plataformas existentes, el tercer capítulo contiene la información de las plataformas de desarrollo de aplicaciones para dispositivos móviles inteligentes a comparar, los parámetros de medición, el análisis y la

comprobación de la hipótesis. El cuarto capítulo tiene el desarrollo del Sistema de Recolección de pedidos.

CAPITULO I

MARCO REFERENCIAL

1.1 TÍTULO DEL PROYECTO DE TESIS:

ANÁLISIS COMPARATIVO ENTRE LAS PLATAFORMAS DE DESARROLLO DE APLICACIONES MÓVILES PARA LOS SISTEMAS OPERATIVOS ANDROID Y IOS.

1.2 LUGAR DE REALIZACIÓN

Bonny Restaurant.

1.3 ANTECEDENTES

El desarrollo de aplicaciones móviles inicialmente tuvo grandes barreras relacionadas a la velocidad de procesamiento y a las limitaciones técnicas de su hardware, mismas que restringían la implementación e instalación de aplicaciones de mediana complejidad.

Hace algunos años estos dispositivos se usaban básicamente para realizar llamadas, escribir mensajes de texto (*SMS*) y entretener al usuario con el uso de juegos sencillos.

Gracias a la evolución del hardware de los dispositivos móviles, su rango de uso se ha ampliado a varios campos, como el multimedia, el uso de sensores, el consumo de servicios web y la navegación por *Internet*. Estas capacidades han generado una demanda de aplicaciones que permitan hacer uso de ellas de manera sencilla e intuitiva, llevando así a los dispositivos móviles a cubrir necesidades que antes sólo lo hacía un computador.

La aparición de los dispositivos móviles inteligentes ha disparado una revolución en los últimos años en lo que concierne a la utilización y concepción de *Internet*.

Hoy en día es amplia la utilización de dispositivos móviles inteligentes y su adecuado uso ha permitido optimizar la gestión de los tiempos de trabajo e incrementar la productividad, todo ello ayudado de una gran portabilidad, característica fundamental de los dispositivos móviles, que ha permitido que ellos sean parte de nuestra vida diaria.

Nuestro país no ha empezado aún por explotar en gran medida esta tecnología, es por ello que no se ha percibido el gran ahorro de tiempo y comodidad que puede ofrecer en varias áreas de acción, incluyendo las de atención al cliente.

Respecto al movimiento de la participación en el mercado mundial de los sistemas operativos para dispositivos móviles a febrero del 2011, según el informe de StatCounter, **Android** se lleva el 15,2% a la alza, frente al 14,5% de **RIM** a la baja, **Symbian** con el 30,7% a la baja, seguida de cerca por **iOS** con el 24,6%. El alza proyectada que presenta Android dentro de estas estadísticas es elevada y llevaría a pensar que es un sistema operativo que a corto plazo ocupará una cuota de gran importancia en el mercado. Este fenómeno ha comenzado a brotar en Estados Unidos donde, las estadísticas según el informe de Nielsen, indican que Android ocupa el 31% de participación durante

el periodo de **Enero del 2011 a Marzo del 2011**. De la misma manera, iOS está manteniendo gran acogida de los usuarios, tal es el caso de Europa, que según el estudio de StatCounter recoge el 43% de mercado de sistemas operativos para dispositivos móviles. Es por este motivo que se puede percibir una tendencia actual de una gran cantidad de aplicaciones móviles desarrolladas para las plataformas android y iOS. Así como las aplicaciones, las herramientas para el desarrollo de las mismas han ido creciendo casi a la par, mejorando la experiencia que brindan a los desarrolladores e incluyendo paulatinamente las nuevas versiones de APIS para estos sistemas operativos.

Esta diversidad de sistemas operativos y plataformas de desarrollo existentes permiten mantener un alto grado de competencia en el mercado, lo cual es bueno para los consumidores, sin embargo es de gran impacto para las personas y empresas que dedican sus esfuerzos a la implementación de aplicaciones y servicios para dispositivos móviles. Para estos últimos, es muy compleja la tarea de escoger una plataforma sobre la cual puedan realizar el desarrollo de un producto software obteniendo una máxima productividad en la implementación, adaptándolo cada vez a las necesidades cambiantes de los clientes. Ante el conjunto de situaciones presentado anteriormente, se plantea el siguiente problema:

¿Qué plataforma de desarrollo para dispositivos móviles inteligentes permite mayor productividad en la implementación de aplicaciones?

Los dispositivos móviles en Ecuador aún no tienen una aplicación en el ámbito de atención al cliente, en donde las colas de espera y la experiencia de las personas puedan ser mejoradas, con un beneficio directo no sólo para el consumidor, sino también para los dueños del establecimiento prestador del servicio.

En los restaurantes del país los clientes usualmente tienen que abrir la carta tradicional para seleccionar su orden y respetar con paciencia características “estándar” de atención al cliente, como las siguientes:

- Realizar su pedido de una manera tradicional hacia un mesero que se acerque a solicitar la orden.
- Existencia de una cola de personas para realizar su pedido.
- Existencia de una cola de clientes para realizar su pago.
- Inexistencia de servicio para la reservación de mesas.
- Inexistencia de información de la disponibilidad de los platos.
- Inexistencia de información detallada acerca de las comidas disponibles (foto, descripción, especies, etc.).

Características adicionales para los dueños del local

- No se lleva estadísticas de los platos más vendidos.
- No existe una forma de recolección de comentarios o sugerencias de los usuarios a cerca de su experiencia en el uso de ese servicio por lo cual no existe un control de quejas o insatisfacción.

Características indirectas

- Falta de motivación para que el cliente acceda al servicio.

En este contexto no existe una buena estrategia para motivar al cliente a que acceda al servicio de restaurante.

En este ámbito, con la ayuda de los dispositivos móviles de bajo costo y el uso de tecnología de comunicación inalámbrica, se puede desarrollar una aplicación software que ayude a este proceso de atención al cliente.

1.4 JUSTIFICACIÓN

1.4.1 Justificación Teórica

Al conocer con exactitud los resultados producidos por este proyecto permitirá conocer que ofrecen cada una de las plataformas de desarrollo de aplicaciones móviles en los sistemas operativos android y iOS.

También permitirá mantener un alto grado de competencia en el mercado, lo cual es bueno para los desarrolladores y empresas que dedican sus esfuerzos a la implementación de aplicaciones y servicios para dispositivos móviles podrán escoger una plataforma sobre la cual puedan realizar el desarrollo de un producto software obteniendo una máxima productividad en la implementación de una aplicación, con la capacidad de adaptarlo a las necesidades cambiantes de los clientes consumidores.

Los procesos más relevantes que se realizarán para el análisis comparativo se realizaran mediante pruebas en prototipos en las dos plataformas android y iOS.

1.4.2 Justificación Aplicativa

Mediante la utilización de dispositivos móviles inteligentes que son accesibles en costos y una utilización inalámbrica, se va a desarrollar una aplicación software para la atención al cliente en Bonny Restaurant que presentará las siguientes características:

Administración:

- Gestión de Clientes.
- Gestión de platos o comidas.
- Gestión de pedidos.
- Gestión de mesas.
- Gestión de Usuarios
- Informes estadísticos de los platos más vendidos en el día.

La aplicación para la atención al cliente va presentar las siguientes características:

- Navegación sobre la carta con información visual (foto, precio, descripción) de las comidas que se ofrece al cliente.
- Solicitud de pedidos.
- Reservación de mesas.

- Autenticación de Usuarios.

La aplicación para presentar la información actualizada va a consumir servicios web de forma inalámbrica que van a estar publicados en un servidor apache, el intercambio de datos se la va a realizar mediante JSON (Figura I - 1).

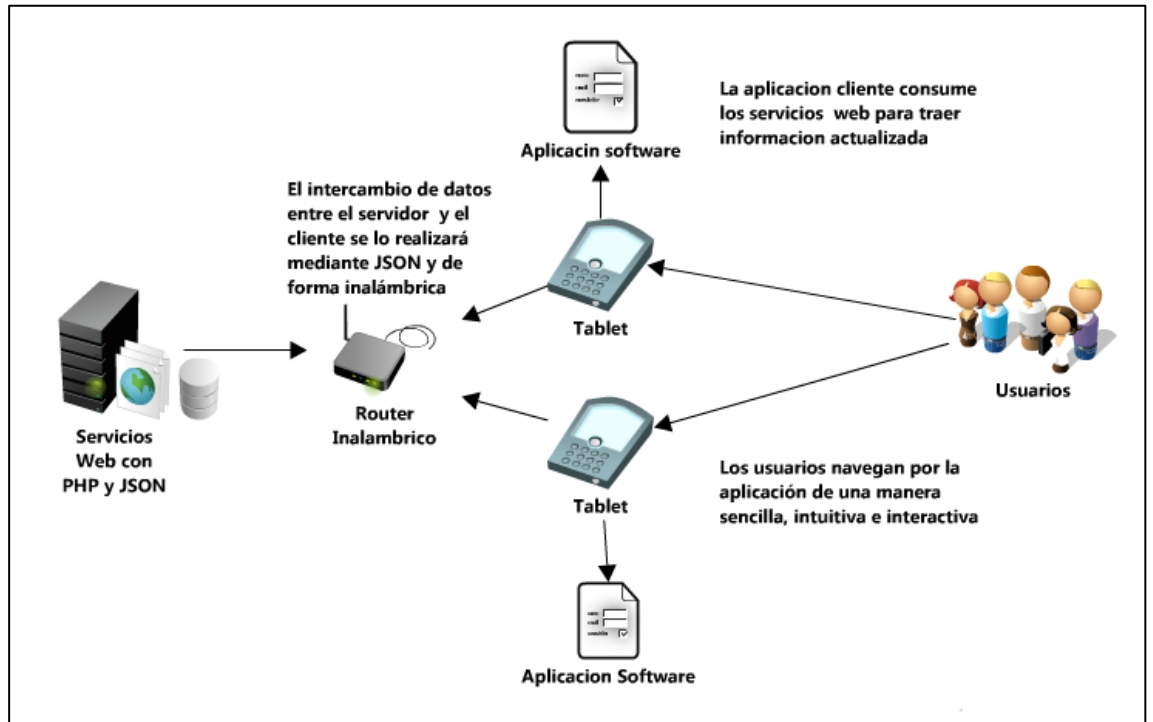


Figura I - 1. Arquitectura de la aplicación

La aplicación a realizar ayudará a mejorar el servicio y ofrecer una nueva experiencia a los clientes en los restaurantes y será implementado en Bonny Restaurant, obteniendo las siguientes ventajas:

- Incrementación de ventas para este sector comercial.
- Mejor experiencia del consumidor al hacer uso de este servicio, ya que los consumidores pueden navegar por las páginas virtuales del menú, elegir de forma divertida su comida observando imágenes y leyendo las características de cada plato.
- Capacidad para registrar el número de platos vendidos, esto permite que los clientes no elijan una comida o bebida que se ha terminado.

1.5 OBJETIVOS

1.5.1 Objetivo General

Realizar un análisis comparativo entre las plataformas de desarrollo de aplicaciones móviles para los sistemas operativos android y iOS.

1.5.2 Objetivos Específicos

- Estudiar las plataformas de desarrollo para dispositivos móviles inteligentes.
- Analizar la plataforma de desarrollo de aplicaciones para los sistemas operativos android y iOS.
- Determinar los parámetros de comparación para obtener una mayor productividad en el desarrollo de aplicaciones para los sistemas operativos android y iOS.
- Seleccionar la plataforma de desarrollo en base a los parámetros de comparación establecidos y a pruebas realizadas en los prototipos.
- Diseñar una aplicación software en base a la plataforma seleccionada, para la atención de clientes en la recolección de pedidos en Bonny Restaurant.

1.6 HIPÓTESIS

Una plataforma de desarrollo para dispositivos móviles inteligentes sobre el sistema operativo android permitirá una mayor productividad que en el sistema operativo iOS en la implementación de aplicaciones.

CAPITULO II

INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES INTELIGENTES

2.1 INTRODUCCION

Se denomina teléfono inteligente o smartphone, en inglés, a un dispositivo que partiendo de las funciones básicas de un teléfono móvil ha evolucionado hasta tener las mismas o más funcionalidades que un computador personal, Casi todos los dispositivos móviles inteligentes disponen de agenda, gestor de correo electrónico, reproductor de música y video, además de infinitas posibilidades de conectividad y funcionalidades o aplicaciones de todo tipo.

Una característica importante de casi todos los dispositivos móviles inteligentes es que permiten la instalación de nuevos programas o aplicaciones. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. El término inteligente hace referencia a cualquier interfaz, como un teclado en miniatura, una pantalla táctil o simplemente el sistema operativo móvil que posee, diferenciando su uso mediante una exclusiva disposición de los menús, teclas, atajos, etc. Entre otras características comunes está la función multitarea, el acceso a Internet vía WiFi o 3G, a los programas de agenda, a una cámara digital integrada, administración de contactos, acelerómetros, GPS y algunos programas de navegación así como la capacidad de leer y editar documentos en formatos como PDF y Microsoft Office.



Figura II - 1. HTC Desire, el smartphone más popular de los teléfonos móviles Android.



Figura II - 2. Iphone, el smartphone de Apple.

2.2 HISTORIA DE LOS DISPOSITIVOS MÓVILES INTELIGENTES

A continuación se describe una breve historia de la evolución de los dispositivos móviles inteligentes, desde sus comienzos con equipos monocromáticos hasta la tecnología de hoy en día, capaz de reproducir y capturar video en alta definición, entre muchas otras aplicaciones.

2.2.1 Primeros teléfonos inteligentes

El primer dispositivo que podemos considerar un smartphone fue creado por IBM conjuntamente con la operadora BELLSOUTH en el año 1992. Simon fue diseñado como una prueba de concepto para mostrar en la feria COMDEX y fue comercializado un año más tarde, en 1993. Se trata del primer dispositivo en el que se implementaron funcionalidades extra a las de un teléfono inalámbrico normal, incluyendo mail fax, calendario, calculadora e incluso un lector de tarjetas PCMCIA.



Figura II - 3. IBM Simon (1992).

Nokia lanzó su primer dispositivo inteligente en 1996, en concreto, su modelo Nokia 9000. Básicamente era una fusión entre las funcionalidades de una PDA y las características de un teléfono inalámbrico normal. Serían otros modelos de este fabricante los primeros en incluir ciertas características indispensables hoy en día para que un dispositivo sea considerado un smartphone, como pantalla a color e inclusión de conectividad WIFI entre otros. Cabe destacar el modelo Nokia 9210 Communicator, lanzado en 1998, por ser el primer dispositivo en adoptar el sistema operativo SymbianOS.

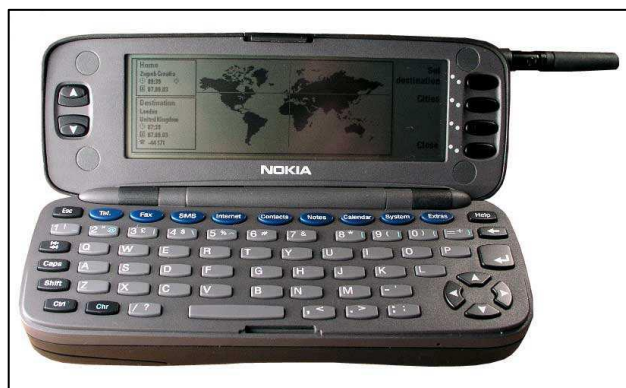


Figura II - 4. Nokia 9000 (1996).

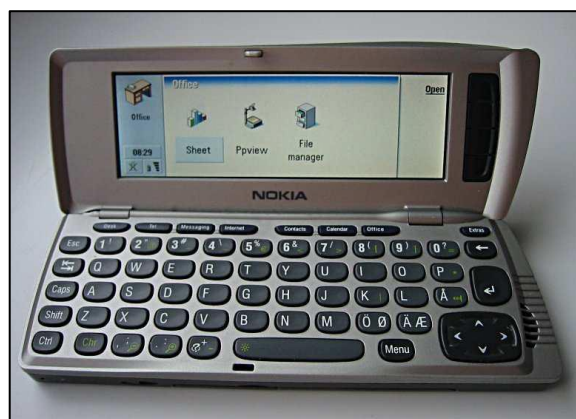


Figura II - 5. Nokia 9210 con pantalla a color (2000).

En el año 1997, Ericsson sacó al mercado su teléfono GS88, que fue el primer dispositivo etiquetado expresamente como smartphone. En modelos posteriores se incluiría pantalla táctil.



Figura II - 6. GS88 de Ericsson (1997).

A partir del año 2000, se produce un incremento significativo en cuanto a la comercialización de distintos modelos de Smartphone. Los hitos más significativos son la salida al mercado de dispositivos Windows CE como ordenador de bolsillo y especialmente la comercialización del primer modelo de Blackberry con funcionalidades de Smartphone en el año 2002 por parte del fabricante RIM (Research in Motion). Este modelo sería el primero en estar optimizado para la gestión del correo electrónico.



Figura II - 7. BlackBerry Quark con pantalla monocromo.(2000)



Figura II - 8. BlackBerry Bold 9900, con teclado y pantalla táctil (2011)

Durante estos años, todos los dispositivos de los diferentes fabricantes, han ido evolucionando e introduciendo nuevas características, convirtiendo la mayoría de funcionalidades como estándar en los productos base (conectividad, pantalla táctil, aplicaciones de gestión y multimedia, etc.), sentando la base de lo que debería ser un teléfono inteligente.

Nacimiento de iPhone y Android

En el año 2007 Apple Inc., introduce su primera generación de dispositivos iPhone, que sería uno de los primeros smartphones que permitían manipularse íntegramente desde su pantalla táctil. El iPhone se lanzó con gran aceptación del público, en Estados Unidos, el 29 de junio de 2007 a las 18:00 (hora oeste), únicamente en las tiendas Apple, tiendas de AT&T y Apple Store (en línea) al precio de 499 dólares la versión de 4 GB y 599 dólares la versión de 8 GB. Este dispositivo marcaría un punto de inflexión en este segmento de mercado. Durante estos últimos años, Apple ha sacado nuevas versiones de su iPhone con soporte 3G y descarga de aplicaciones desde su propia comunidad denominada App Store.



Figura II - 9. Lanzamiento del primer Iphone en EEUU (2007)

En el 2008 sale a la luz Android, una plataforma de código abierto basada en una modificación del kernel de Linux y específico para Smartphones. Se convierte en estandarte del consorcio Open Handset Alliance, creado e impulsado por Google en el año 2007 y compuesto por diversos fabricantes, desarrolladores y operadores (Intel, HTC, Dell, ARM, Motorola, entre otros) con el objetivo de desarrollar estándares abiertos para dispositivos móviles.



Figura II - 10. El primer smartphone con Android: HTC Dream (Octubre 2008)

El primer dispositivo en utilizar la plataforma Android como base fue el HTC Dream, distribuido por T-Mobile como G1. El software implementado incluía la integración de las aplicaciones de Google (Maps, Calendar, Gmail, y el navegador Chrome). Como novedad, destaca el uso de aplicaciones de terceras partes (gratuitas y de pago) mediante la comunidad Google Play antes llamado Android Market. Otros fabricantes han seguido esta línea de crear una comunidad propia para la gestión de aplicaciones externas: RIM y su BlackBerry App World, Nokia con su Ovi Store (Mayo 2009), Palm y Palm App Catalog (Junio 2009) y Microsoft con Windows Marketplace for Mobile (Octubre 2009).

2.3 MERCADO DE LOS DISPOSITIVOS MÓVILES INTELIGENTES

En un segmento competido, como es el de los dispositivos móviles inteligentes, las cifras de ventas, la participación de mercado y conocer los comportamientos de los consumidores es importante para ser competitivo y posicionarse. Los grandes competidores: Samsung, Apple, BlackBerry, LG, Nokia, HTC, entre otros, afirman que hay mucho por ganar, en un segmento que está en constante crecimiento.

El más reciente estudio de Vision Mobile¹, en el que analiza el estado del mercado de los dispositivos móviles inteligentes en el mundo durante el primer semestre del año, como se muestra en el siguiente gráfico:

¹ www.visionmobile.com

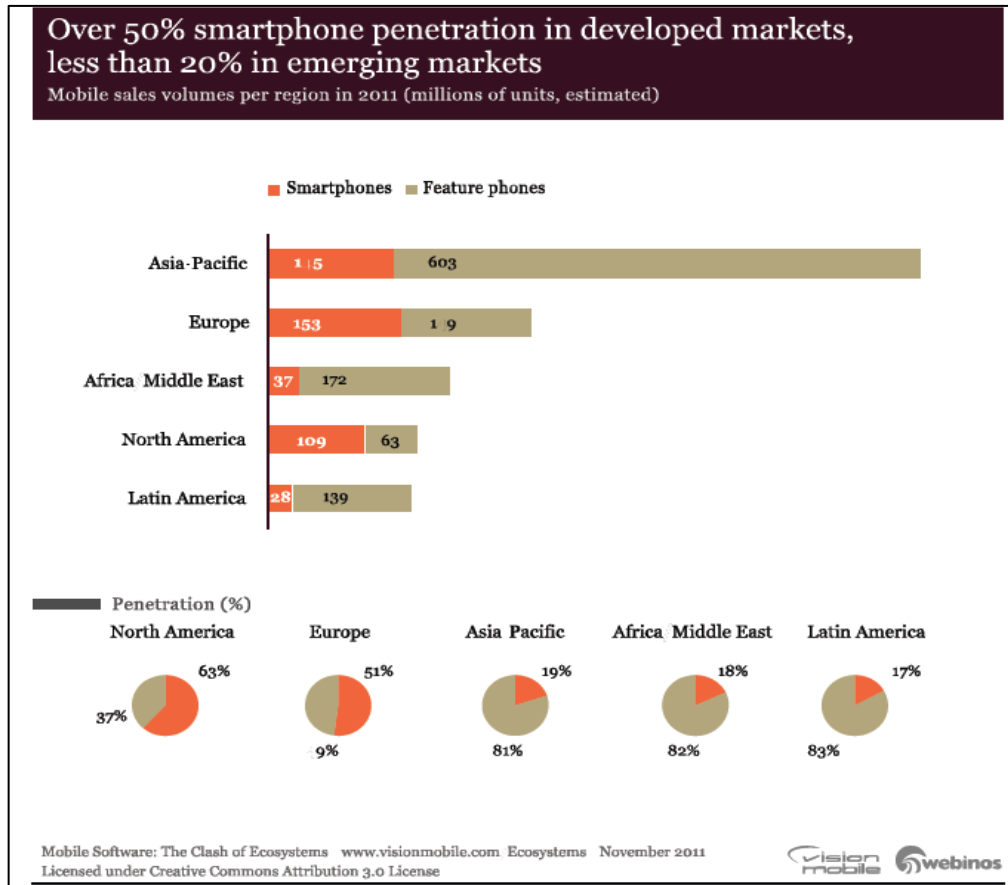


Figura II - 11. Estudio de Vision Mobile

Podemos observar que América Latina con un 17%, es una de las regiones con menor penetración de este tipo de tecnología, lo que a su vez la posiciona como un gran mercado para explorar. La cifra de América Latina del 17% está lejos de la media de penetración de los teléfonos inteligentes en el mundo, que según el estudio es del 27%.

También podemos observar el nivel de penetración en las diferentes regiones:

Estados Unidos es de 63%

Europa es de 51%

Asia es de 19%

África y Oriente Medio es de 18%

América del Sur, es de 17%.

Además, se estima que para este mismo periodo, en el 2012, la mitad de los dispositivos móviles serán dispositivos móviles inteligentes, y en 2013 sean la mayoría.

La penetración de móviles a nivel mundial, que es del 27%, la lideran Apple y Nokia, cada uno con un 5%, seguido por Samsung con el 4%, mientras que RIM y HTC alcanzan respectivamente un 3%. LG, Motorota, Huawei, Sony Ericcson, ZTE y otros teléfonos inteligentes, tiene en cada caso un 1% del mercado.

En relación con los sistemas operativos, que dominan en los dispositivos móviles inteligentes que hay en el mercado mundial, ese 27%, se divide de la siguiente manera:

Android 11%

Symbian OS 5%,

IOS 5%

BlackBerry OS 3%

Windows Phone 1%

otros sistemas 1%.

En la siguiente tabla se muestra el número de aplicaciones que existen en cada una de las plataformas hasta la fecha.

TABLA II - I **Número de aplicaciones por plataforma**

Plataforma	Numero de Aplicaciones
iOS	500,000
Android	300,000
BlackBerry OS	15,000
Symbian	25,000
Windows Phone	30,000
webOS	7,000

Fuente: Vision Mobile

Se puede observar que el proveedor más grande en tiendas de aplicaciones es App Store, de la corporación Apple, lanzado en 2008. El crecimiento en el

mercado de aplicaciones para dispositivos móviles inteligentes se debe también a la gran aparición de la tienda en línea Google Play antes llamada Android Market, que hoy en día está cerca de la tienda de App Store de Apple, la cual cuenta con miles de aplicaciones no gratuitas y gratuitas.

En cuanto al porcentaje de desarrolladores que existen en las plataformas más populares, se puede observar en la siguiente tabla:

TABLA II - II **Porcentaje de desarrolladores por plataforma**

Plataforma	% de desarrolladores usando la plataforma
Android	67%
iOS	59%
Windows Phone 7	36%

Fuente: Vision Mobile

En el estudio de Vision Mobile para realizar el informe, se habría tenido en cuenta a personal técnico de 75 países diferentes. Actualmente, un 67% de los desarrolladores utilizaría la plataforma de Google, frente a un 59% que usa Apple. Los desarrolladores móviles empiezan a posicionarse a favor del sistema operativo móvil Android frente a los del resto. Mientras que Windows Phone 7 todavía no sería visto por los desarrolladores como una plataforma viable comercialmente. La plataforma de Nokia sería, junto a Java ME, la que contaría con mayor tasa de abandono por parte de programadores si se comparan los porcentajes con los registrados un año atrás.

2.4 SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES INTELIGENTES

Según los datos de ventas de terminales actuales, los sistemas operativos que dominan el mercado son Android, iOS, Windows Phone 7, BlackBerry OS y

Symbian OS. Además de estas plataformas dominantes, existen otros sistemas minoritarios como es el caso de MeeGo.

A continuación, se describirán las características principales de las plataformas mencionadas.

2.4.1 BlackBerry OS

El sistema operativo BlackBerry OS², nace de la mano de la empresa canadiense RIM (Research In Motion)³ y comparte nombre con su línea de teléfonos inteligentes lanzada al mercado en 1999.

Las principales características de este sistema operativo se centran en el mercado de los negocios. Como ejemplo podemos citar que ha sido una de las primeras plataformas en proveer correo electrónico en tiempo real.

Con la llegada de otros sistemas operativos para teléfonos inteligentes, BlackBerry OS añade mejores capacidades multimedia, soporte para MIDP (perfiles informativos del terminal), pantallas táctiles y el resto de elementos que, en general, podemos encontrar en sus competidores.

En la actualidad una de las características que hacen a este sistema operativo único reside en que los servicios de datos pasan por los servidores de RIM, permitiendo así la compresión de la información que, además, viaja cifrada por la red, obteniendo velocidad y seguridad.

A pesar de ser uno de los principales líderes del mercado, BlackBerry OS no ha sabido adelantar a su competencia en sectores alejados de los negocios, es uno de los pocos sistemas operativos dominantes que no hace uso de la tecnología multitáctil⁴.

² http://es.wikipedia.org/wiki/BlackBerry_OS

³ <http://www.rim.com/>

⁴ http://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil

Como plataforma de desarrollo se encuentra lejos de Android e iOS. Prueba de ello es la escasa cantidad de aplicaciones disponibles y de aplicaciones que sus usuarios instalan.

2.4.1.1 Desarrollando para RIM

RIM provee un sistema operativo propietario, multitarea para los dispositivos BlackBerrys que hace uso intensivo de los dispositivos de entrada como el teclado y el trackball⁵. El sistema operativo provee soporte para MIDP⁶ 1.0 y 2.0 así como para WAP⁷ 1.2.

Cualquier desarrollador puede escribir software usando las APIs⁸ propietarias de BlackBerry, pero para que una aplicación pueda acceder a ciertas funcionalidades restringidas debe estar firmado digitalmente con un certificado asociado a una cuenta de desarrollador en RIM, este proceso garantiza la procedencia de las aplicaciones, no así la calidad del código.

2.4.2 Symbian

Symbian⁹ es el sistema operativo móvil que más éxito comercial ha tenido si atendemos a la cantidad de dispositivos que se han vendido.

En sus orígenes, Symbian es controlado por varias empresas, entre todas intentan cooperar para avanzar rápidamente hacia el entonces incipiente mundo de los teléfonos inteligentes, pero con el paso de los años es la compañía sueca Nokia¹⁰ la principal encargada del desarrollo del sistema operativo.

⁵ <http://es.wikipedia.org/wiki/Trackball>

⁶ <http://es.wikipedia.org/wiki/MIDP>

⁷ http://es.wikipedia.org/wiki/Wireless_Application_Protocol

⁸ http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

⁹ <http://www.symbian.org/>

¹⁰ <http://www.nokia.es/>

A pesar de su despegue y años de mantenimiento, va a ser la filosofía de Nokia la que impida a Symbian seguir creciendo y consolidarse como un sistema operativo móvil dominante. Las actualizaciones de sistema tardan en trasladarse a los usuarios, incluso a los poseedores de teléfonos de gama alta, por lo que poco a poco los más interesados en sacar partido a su teléfono han ido migrando a otras opciones, si bien es cierto que la gama media de teléfonos con Symbian de Nokia sigue siendo bastante popular.

Estos vaivenes afectan a los desarrolladores, que han visto como en casi cuatro años no salía a la luz una revisión importante del sistema operativo, por lo que muchas de las opciones que existen en otras plataformas no están disponibles.

El éxito de ventas de iPhone y el rápido desarrollo de la plataforma Android ha hecho que Nokia libere el código de Symbian para tratar de hacerlo avanzar más rápido gracias a la comunidad. Aun no hay datos para poder comprobar si esta solución ha sido acertada ya que el primer terminal con Symbian3, la última versión del mismo, acaba de ser lanzada al mercado.

2.4.2.1 Desarrollando en Symbian OS

El lenguaje nativo de Symbian OS es el C++ aunque no en una implementación estándar. Existen múltiples SDKs (Software Development Kit)¹¹ para el desarrollo de aplicaciones. Algunos fabricantes ofrecen SDKs propios o extensiones a los SDK para sus productos o para familias de productos que se pueden bajar de los sitios web.

Los SDK oficiales contienen documentación, los headers, las librerías necesarias para compilar un software Symbian, emuladores basados en Windows y un compilador. Hasta la versión 8 se incluye como compilador

¹¹ http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software

GCC¹², la versión 9 usa una nueva ABI¹³ (application binary interface) y requiere un compilador distinto.

La programación en C++ para Symbian requiere el uso de técnicas especiales como descriptores o CleanupStack¹⁴, esto puede hacer que programas relativamente simples sean más difíciles de implementar que en otros entornos. Actualmente las técnicas de programación necesarias para desarrollar en Symbian hacen que los programas sean propensos a errores en rutinas de bajo nivel en lugar de errores en las funcionalidades específicas de la aplicación.

El primer IDE oficial y comercial para Symbian, Codewarrior, fue reemplazado durante el 2006 por Carbide c++¹⁵ un IDE basado en Eclipse desarrollado por Nokia que se ofrece en tres versiones:

- Carbide.c++ OEM Edition for device creation users.
- Carbide.c++ Professional Edition for developers working with preproduction devices.
- Carbide.c++ Developer Edition for application development on production phones.

Muchos de los dispositivos con Symbian OS además de C++ pueden ser programados en OPL¹⁶, Python, Visual Basic, Simkin y Perl, así como en Java ME.

El plugin para Visual Studio de AppForge llamado "AppForge CrossFire" que permitía programar en un dispositivo Symbian en Visual Basic, VB.NET o C# dejó de estar disponible en el 2007 cuando Oracle compró la propiedad intelectual de AppForge y anunció que no tiene planeado vender o prestar soporte a los viejos productos de esta empresa. Finalmente también existe una versión del IDE de Borland para Symbian, así como herramientas o técnicas desarrolladas por la comunidad para programar en Linux o MacOS.

¹² http://es.wikipedia.org/wiki/GNU_Compiler_Collection

¹³ http://en.wikipedia.org/wiki/Application_binary_interface

¹⁴ http://en.wikipedia.org/wiki/Cleanup_stack

¹⁵ <http://es.wikipedia.org/wiki/Carbide.c%2B%2B>

¹⁶ http://en.wikipedia.org/wiki/Open_Programming_Language

2.4.3 Windows Phone 7

Windows Phone¹⁷ ha sido tan solo uno de los nombres que ha tenido el sistema operativo para móviles de Microsoft¹⁸, antes fue conocido como PocketPC y Windows Mobile.

A comienzos de la década se disputaba el título de sistema operativo más popular con Symbian. Sin embargo ambos líderes del mercado han visto como con la llegada de nuevos dispositivos con infinidad de opciones y atractivos diseños han mermado su cuota de mercado.

El desarrollo de la plataforma de Microsoft comienza con la venta de PDAs¹⁹ que no tenían servicios de telefonía asociado, cubriendo la necesidad de ejecutar aplicaciones parecidas a las de un ordenador personal en un aparato más ligero. Como consecuencia de lo anterior la interfaz del sistema trata de copiar la interfaz tradicional del sistema operativo Windows en dispositivos mucho más pequeños. No es hasta las versiones más recientes de Windows Mobile cuando la interfaz comienza a evolucionar en el mismo sentido que su competencia. El hecho de que a finales de 2010 Windows Mobile no tenga soporte para pantallas multitáctiles deja claro lo atrasado del sistema.

En febrero de 2010 Microsoft anuncia que a finales de año estará disponible Windows Phone 7, diseñado con la idea de distinguirse de la competencia. La crítica ha hablado bien de la interfaz gráfica aunque hay aspectos de un sistema operativo moderno que faltaran en esta versión. Las características omitidas más importantes son, además de la capacidad multitáctil citada anteriormente:

- Soporte para tarjetas de memoria.
- Soporte para multitarea.

¹⁷ <http://www.microsoft.com/windowsphone/es-es/default.aspx>

¹⁸ <http://www.microsoft.com/es/es/default.aspx>

¹⁹ <http://es.wikipedia.org/wiki/PDA>

- Soporte para Flash.
- Soporte para copiar y pegar.
- Soporte para anclaje a red.

2.4.3.1 Desarrollando en Windows Phone 7

El desarrollo de aplicaciones para Windows Phone 7 puede hacerse empleando dos tipos de implementaciones:

Microsoft Silverlight²⁰ que permite realizar aplicaciones que contengan transiciones y efectos visuales. Silverlight permite el desarrollo de aplicaciones basadas en XAML²¹. Silverlight para Windows Phone incluye el Microsoft .NET Compact Framework, que hereda de la arquitectura .NET Framework²², el CLR y la ejecución de código administrado, soporta un subconjunto de las librerías de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Este soporte incluye el Base Class Library, una colección de clases que soportan lectura y escritura de ficheros, manipulación XML y manejo de gráficos. Cada aplicación que es ejecutada en Windows Phone OS 7.0 se ejecuta dentro de un proceso en el motor de ejecución .NET Compact Framework

Microsoft XNA Framework²³ es una implementación nativa de .NET Compact Framework que incluye un amplio conjunto de bibliotecas de clases, específicos para el desarrollo de juegos, por ejemplo para el manejo de dispositivos de entrada, tratamiento de sonidos y vídeos, carga de modelos y texturas, uso de ficheros de forma transparente a la plataforma en la que se ejecute, desarrollo de juegos online, etc. Permite desarrollar juegos para Windows Phone OS 7.0 , Xbox 360²⁴ y Windows 7

²⁰ es.wikipedia.org/wiki/Microsoft_Silverlight

²¹ <http://es.wikipedia.org/wiki/XAML>

²² http://es.wikipedia.org/wiki/Microsoft_.NET

²³ http://es.wikipedia.org/wiki/Microsoft_XNA

²⁴ http://es.wikipedia.org/wiki/Xbox_360

El soporte, ayuda e información para el desarrollo de aplicaciones se realiza desde el Centro de Desarrollo de MSDN en español²⁵.

2.4.4 Android

Según Marko Gargenta (5), android es una plataforma completa de código abierto diseñado para dispositivos móviles. Es promovido por Google y pertenece a Open Handset Alliance. El objetivo de la alianza es "acelerar la innovación en los consumidores móviles y ofrecen una rica, menos costosa, y mejor experiencia móvil". Como tal, Android está revolucionando el espacio móvil. Por primera vez, es una verdadera plataforma abierta que separa el hardware del software que se ejecuta en él. Esto permite que para un número mucho mayor de dispositivos puedan ejecutar las mismas aplicaciones y crea un rico ecosistema de desarrolladores y consumidores.

Andy Rubin de Google, android se describe como, la primera plataforma verdaderamente abierta y completa para dispositivos móviles, tiene todo el software para ejecutarse en un teléfono móvil pero sin los obstáculos de propiedad que tiene la innovación móvil²⁶.

Los dos autores coinciden en que, Android es una combinación de tres componentes:

- Un sistema libre, operativo de fuente abierta para dispositivos móviles.
- Una plataforma de desarrollo open-source para la creación de aplicaciones para dispositivos móviles, especialmente los dispositivos móviles, que ejecutan el sistema operativo Android
- Aplicaciones de creadas para el sistema operativo.

²⁵ http://es.wikipedia.org/wiki/Microsoft_Developer_Network

²⁶ <http://googleblog.blogspot.com/2007/11/Wheres-mi-gphone.html>

Más concretamente, Android se compone de varias piezas necesarias y dependientes, incluyendo las siguientes:

- Un diseño de referencia de hardware que describe las capacidades necesarias para un dispositivo móvil para apoyo a la pila de software.
- Un sistema operativo Linux kernel que proporciona una interfaz de bajo nivel con el hardware, la gestión de memoria y control de procesos, todos optimizados para dispositivos móviles.
- Bibliotecas de código abierto para desarrollo de aplicaciones, incluyendo SQLite, WebKit, OpenGL, y un encargado de los medios.
- Un tiempo de ejecución utilizado para ejecutar y alojar aplicaciones de Android, incluido la máquina virtual Dalvik y las bibliotecas del núcleo que proporcionan funcionalidad específica en Android. El tiempo de ejecución es diseñado para ser pequeño y eficiente para el uso en dispositivos móviles.
- Un framework de aplicación que expone los servicios del sistema de la capa de aplicación, incluyendo el gestor de ventanas y gerente de locación, los proveedores de contenidos, la telefonía, y sensores
- Un marco de interfaz de usuario que se utiliza para alojar y ejecutar aplicaciones.
- Aplicaciones preinstaladas enviado como parte de la pila.
- Un kit de desarrollo de software utilizado para crear aplicaciones, incluyendo herramientas, plug-ins, y la documentación.

2.4.4.1 Desarrollando para Android

Se puede desarrollar una aplicación Android con las mismas herramientas utilizadas para desarrollar en Java. Las "Android core libraries" proveen las funcionalidades necesarias para construir impresionantes y poderosas aplicaciones para dispositivos móviles.

El "Android SDK" soporta Windows XP, Windows Vista, Linux y Mac OS y utiliza como IDE de desarrollo a Eclipse mediante el plug in "Android Development Tools (ADT)" aunque es posible utilizar Apache Ant²⁷ para entornos Mac o Linux

2.4.5 iOS

Según Dr, Rory Lewis (3), iOS (anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone, siendo después usado en el iPod Touch e iPad. Es un derivado de Mac OS X, que a su vez está basado en Darwin BSD. El iOS tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios de comunicación" y la capa de "Cocoa Touch". Todo el sistema se encuentra en la partición "/root" del dispositivo, ocupa poco menos de 500 megabytes.

Según Wei Meng Lee (4), iOS es el sistema operativo que utiliza el iPod touch y el iPhone, diseñado por 175 ingenieros de Apple, entre los cuales figuran Rafael Pérez, Danel Abreu, entre otros. Está basado en una variante del Mach kernel que se encuentra en Mac OS X.

Los dos autores coinciden en lo siguiente:

- iOS es un derivado de Mac OS X
- es software propietario

²⁷ http://es.wikipedia.org/wiki/Apache_Ant

- Sistema operativo que utiliza el iPhone, el iPod Touch e iPad.

2.4.5.1 Desarrollando para iOS

El sistema operativo del iOS es, en esencia, el mismo que usa Mac²⁸, desde hace casi diez años, un Unix BSD²⁹ súper cargado. El sistema operativo fue desarrollado por NeXT³⁰, empresa que también desarrollo un novedosísimo (para entonces) framework de programación orientado a objetos. Ese framework evolucionó en lo que hoy conocemos como Cocoa³¹, específicamente Cocoa Touch³² en el caso del iPhone, la base sobre la que se construye cualquier aplicación nativa para el iPhone/iPod Touch.

Para que el SDK del iPhone compile código ARM³³ que es la arquitectura del sistema real es necesario firmar la aplicación, lo que requiere la certificación de Apple. Una nueva firma es necesaria para distribuir la aplicación y por supuesto, si queremos que esta esté disponible en el appStore (página oficial de distribución de aplicaciones para el entorno MacOS) deberemos solicitar la aprobación de Apple.

2.4.6 MeeGo

Meego³⁴ es un producto que ha visto la luz como consecuencia de la fusión de otros dos, Maemo³⁵ de Nokia y Moblin³⁶ de Intel. Estos dos sistemas se caracterizaban por un enfoque distinto al de los sistemas mayoritarios y, con su unión, se han visto reforzadas sus posibilidades de conseguir una buena cuota de mercado.

²⁸ <http://es.wikipedia.org/wiki/Mac>

²⁹ <http://es.wikipedia.org/wiki/Unix>

³⁰ <http://es.wikipedia.org/wiki/NeXT>

³¹ http://en.wikipedia.org/wiki/Cocoa_%28API%29

³² http://es.wikipedia.org/wiki/Cocoa_Touch

³³ http://es.wikipedia.org/wiki/Arquitectura_ARM

³⁴ <http://meego.com/>

³⁵ <http://maemo.org/>

³⁶ <http://moblin.org/>

Las aspiraciones de este sistema son las de crear una base con licencia libre sobre la que colocar distintas capas en función del dispositivo en el que vaya a funcionar.

A día de hoy las capas serán:

- Netbook, pensada para portátiles ultraligeros.
- Handset, pensada para terminales móviles.
- Tablet, destinada a tablet PC.
- In-Vehicle Infotainment, cuya misión será el panel de control de automóviles.

La idea de ofrecer distintas capas, llamadas experiencias de usuario, es una idea novedosa ya que hasta el momento tanto iPhone OS como Android parecen querer competir en todos los dispositivos con el mismo producto. Si a esto le añadimos la importante lista de empresas del sector de la automoción que planean trabajar con Meego, como pueden ser BMW, General Motors o Peugeot-Citroen, convierten a esta plataforma en una opción con puntos fuertes muy diferentes de las demás.

Como puntos flacos, cabe destacar la incertidumbre de su futuro, que desarrollar para distintas experiencias de usuario conlleva aprender distintas tecnologías, lo que conlleva duplicar esfuerzos, y que solo se pueda desarrollar desde sistemas GNU/Linux, por lo que se reduce el número de potenciales desarrolladores.

CAPITULO III

ANÁLISIS COMPARATIVO

3.1 INTRODUCCIÓN

La determinación de una plataforma de desarrollo para dispositivos móviles inteligentes que se va a utilizar para la implementación de una aplicación software para la atención de clientes en la recolección de pedidos, debe ser fundamental para la toma de decisiones de desarrolladores y empresas que dedican sus esfuerzos al realizar este tipo de proyectos, esta determinación debe estar basada en un profundo análisis, de acuerdo a parámetros o criterios de comparación.

3.2 DETERMINAR LAS PLATAFORMAS DE DESARROLLO PARA DISPOSITIVOS MÓVILES INTELIGENTES A COMPARAR

Hoy en día es amplia la utilización de dispositivos móviles inteligentes y su adecuado uso ha permitido optimizar la gestión de los tiempos de trabajo e incrementar la productividad; en el mercado existe una gran cantidad de plataformas para el desarrollo de estas aplicaciones para estos dispositivos, que son muy populares y de gran importancia para los desarrolladores; de las cuales hemos seleccionado dos plataformas por las siguientes razones:

- En el sitio web de la encuestadora statCounter³⁷ se puede visualizar (Figura III - 1) a nivel mundial: La porción de uso de los distintos sistemas operativos para dispositivos móviles inteligentes esto se basa en la amplia disponibilidad de aplicaciones, costos y estabilidad que existen para cada una de las plataformas, las más destacadas son:

- SymbianOS (32% a la baja)
- Android (20% a la alza)
- IOS (21% a la alza)
- BlackBerry OS (12% a la baja)

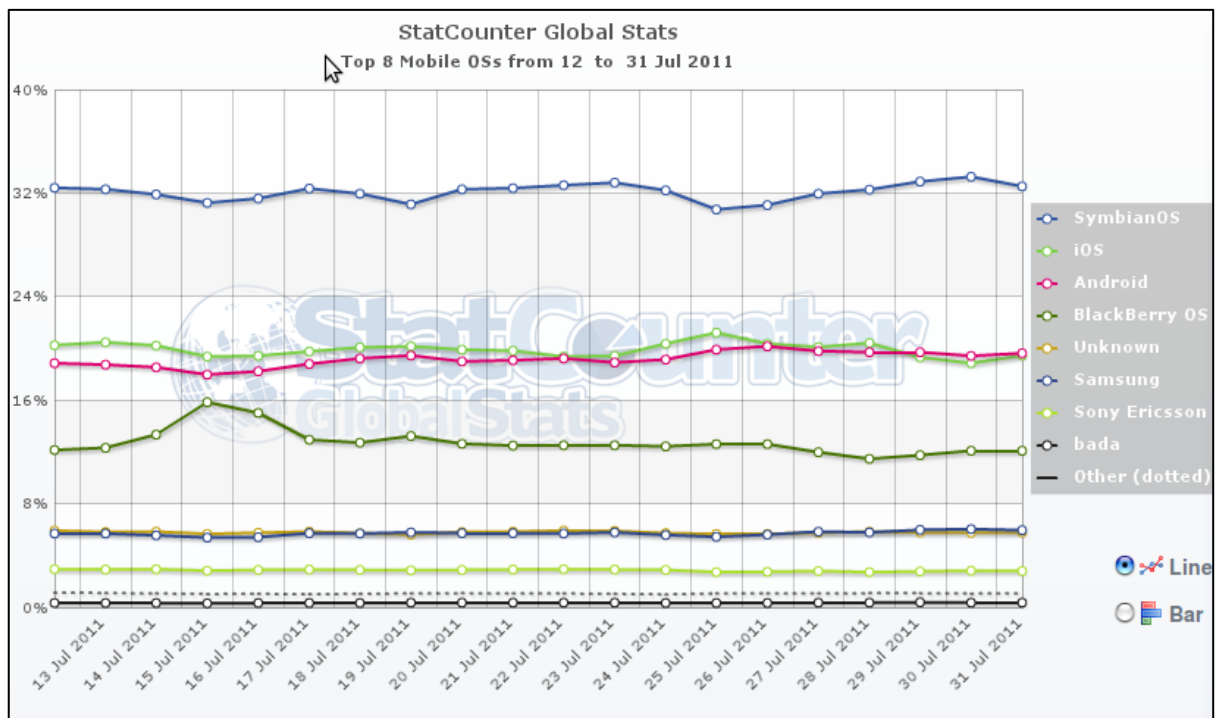


Figura III - 1 Sistemas Operativos más utilizados en Dispositivos Móviles Inteligentes

³⁷ <http://www.maestrosdelweb.com/editorial/cloud-computing-nueva-era-de-desarrollo/>

- En el sitio web de Nielsen³⁸ publica los resultados de sus estudios sobre el mercado de dispositivos móviles inteligentes en Estado Unidos y se destaca el crecimiento de android con un 39%, una diferencia del 11% respecto a iOS, el segundo en la lista de sistemas operativos más utilizados.

Si hacemos la división por fabricantes Apple lidera con un 28% mientras que en Android hay más fragmentación: un 14% para HTC, un 11% Motorola, 8% Samsung y 6% otras compañías (Figura III - 2).

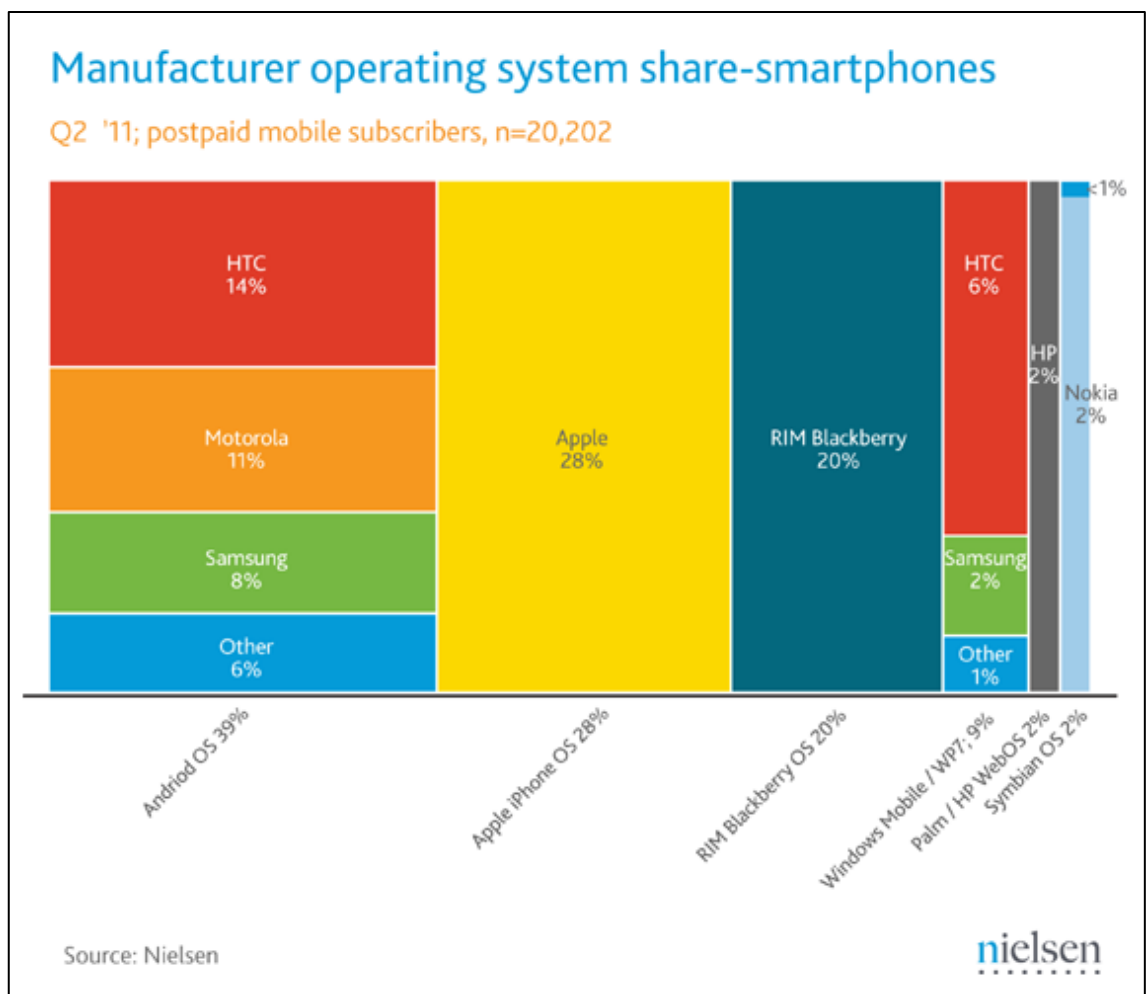


Figura III - 2 Cuota de Mercado de Sistemas Operativos en Estados Unidos

³⁸ http://blog.nielsen.com/nielsenwire/online_mobile/in-u-s-smartphone-market-android-is-top-operating-system-apple-is-top-manufacturer/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+Nielsen+Wire+%28Nielsen+Wire%29

- En la página³⁹ se destaca el porcentaje de descargas de aplicaciones para los sistemas operativos android y iOS. El sistema de Google consigue una cuota del mercado de descargas del 44% y supera a iOS de Apple con 31%, aquí se nota claramente que las aplicaciones se han convertido en uno de los elementos de valor en los distintos sistemas operativos móviles. La batalla entre los principales actores, Android y iOS, se suele desarrollar en el número de aplicaciones y en el volumen de descargas. Según ABI Research⁴⁰, la base de Android instalada supera a iOS y por cada 2,4 terminales con SO de Google hay un iPhone en todo el mundo. Las previsiones de la compañía apuntan a que en 2016 la diferencia aumente y se sitúe en un iPhone por cada tres Android.

Realizando un estudio en estos sitios visitados, encuestadoras y a otros sitios consultados y no mencionados se han seleccionado dos plataformas para la realización de este análisis comparativo: Android de Google y iOS de Apple, las mismas que tienen una gran cuota de mercado, y aceptación entre los consumidores y desarrolladores, y que además cuentan con un conjunto de herramientas y recursos para su utilización.

3.3 ANÁLISIS DE LAS PLATAFORMAS SELECCIONADAS

A continuación realizaremos un análisis detallado de cada una de las plataformas definidas anteriormente:

3.3.1 Android

Android es una tecnología que facilita el trabajo a los desarrolladores, proporcionando una amplia documentación oficial y no oficial.

³⁹ <http://www.europapress.es/portaltic/movilidad/sector/noticia-android-supera-ios-descarga-aplicaciones-20111025171559.html>

⁴⁰ <http://www.abiresearch.com/press/3799>

Android es un Sistema operativo, con el cual funciona cualquier dispositivo móvil inteligente.

Según Marko Gargenta (5), android es una plataforma completa de código abierto diseñado para dispositivos móviles. Es promovido por Google y pertenece a Open Handset Alliance. El objetivo de la alianza es "acelerar la innovación en los consumidores móviles y ofrecen una rica, menos costosa, y mejor experiencia móvil". Como tal, Android está revolucionando el espacio móvil. Por primera vez, es una verdadera plataforma abierta que separa el hardware del software que se ejecuta en él. Esto permite que para un número mucho mayor de dispositivos puedan ejecutar las mismas aplicaciones y crea un rico ecosistema de desarrolladores y consumidores.

Andy Rubin de Google, android se describe como, la primera plataforma verdaderamente abierta y completa para dispositivos móviles, tiene todo el software para ejecutarse en un teléfono móvil pero sin los obstáculos de propiedad que tiene la innovación móvil⁴¹.

Los dos autores coinciden en que, Android es una combinación de tres componentes:

- Un sistema libre, operativo de fuente abierta para dispositivos móviles.
- Una plataforma de desarrollo open-source para la creación de aplicaciones para dispositivos móviles, especialmente los dispositivos móviles, que ejecutan el sistema operativo Android
- Aplicaciones de creadas para el sistema operativo.

Más concretamente, Android se compone de varias piezas necesarias y dependientes, incluyendo las siguientes:

⁴¹ <http://googleblog.blogspot.com/2007/11/Wheres-mi-gphone.html>

- Un diseño de referencia de hardware que describe las capacidades necesarias para un dispositivo móvil para apoyo a la pila de software.
- Un sistema operativo Linux kernel que proporciona una interfaz de bajo nivel con el hardware, la gestión de memoria y control de procesos, todos optimizados para dispositivos móviles.
- Bibliotecas de código abierto para desarrollo de aplicaciones, incluyendo SQLite, WebKit, OpenGL, y un encargado de los medios.
- Un tiempo de ejecución utilizado para ejecutar y alojar aplicaciones de Android, incluido la máquina virtual Dalvik y las bibliotecas del núcleo que proporcionan funcionalidad específica en Android. El tiempo de ejecución es diseñado para ser pequeño y eficiente para el uso en dispositivos móviles.
- Un framework de aplicación que expone los servicios del sistema de la capa de aplicación, incluyendo el gestor de ventanas y gerente de locación, los proveedores de contenidos, la telefonía, y sensores
- Un marco de interfaz de usuario que se utiliza para alojar y ejecutar aplicaciones.
- Aplicaciones preinstaladas enviado como parte de la pila.
- Un kit de desarrollo de software utilizado para crear aplicaciones, incluyendo herramientas, plug-ins, y la documentación.

3.3.1.1 Características

- Amplia variedad de diseños (VGA, librerías de gráficos 2D y 3D).
- Almacenamiento de datos en BBDD SQLite.

- Conectividad (GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth y Wi-Fi).
- Mensajería (SMS y MMS).
- Navegador Web.
- Máquina virtual de Java.
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual de Dalvik, la cual es una especializada máquina virtual diseñada para uso en dispositivos móviles.
- Soporte de formatos (MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF).
- Soporte para hardware adicional (cámaras de video, pantallas táctiles, GPS, acelerómetros).
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE).

3.3.1.2 Restricciones

Al ser Android una plataforma de código abierto no tiene ningún tipo de restricciones para desarrolladores. El único requerimiento para publicar una aplicación es pagar una sola cuota de 25 dólares en Google CheckOut.

3.3.1.3 Versiones de la Plataforma

Existen diferentes versiones de Android que se detallan a continuación:

Android Versión 1.5 (Cupcake)

Lanzamiento: Abril de 2009

Funcionalidades: Teclado QWERTY virtual, widgets, captura de video, copy & paste, subir videos a Youtube en forma directa.

Lo mejor de esta versión: Es ideal para coleccionistas.

Android Versión 1.6 (Donut)

Lanzamiento: Septiembre de 2009

Funcionalidades: Navegación en Google Maps paso-a-paso, mejora en el interfaz de la cámara de foto y video, búsquedas por voz.

Lo mejor de esta versión: muy buena relación coste beneficio.

Android Versión 2.0/2.0 (Eclair)

Lanzamiento: Diciembre de 2009

Funcionalidades: Zoom digital de la cámara de fotos y video, mejoras en Google Maps, nuevo navegador de internet con soporte de video e incluso de Flash, salvapantallas animadas.

Lo mejor de esta versión: muy robusto y estable.

Android Versión 2.2 (Froyo)

Lanzamiento: Mayo de 2010

Funcionalidades: mejora de memoria, mucho más veloz que las versiones anteriores, hotspot WiFi (tethering) incluido, soporte de correo electrónico Microsoft Exchange y video llamada.

Lo mejor de esta versión: muy estable

Android Versión 2.3 (Ginger Bread)

Lanzamiento: Diciembre de 2010

Funcionalidades: Mejora del soporte de video online, mejora de la gestión de la energía (y consumo de la batería), mejora del teclado virtual, soporte para pagos mediante NFC.

Lo mejor de esta versión: abre nuevas posibilidades para usuarios totalmente conectados

Android Versión 3.0/3.4 (Honey Comb)

Lanzamiento: Enero / Mayo 2011

Funcionalidades: Sistema operativo optimizado para tablets, posee plenas funcionalidades multitareas (multitasking) para cambiar de aplicaciones en uso dejando las demás en espera visibles en una columna, soporte Flash y DivX, aparte de los navegadores privados tradicionales integra Dolphin, home page personalizable y widgets.

Lo mejor de esta versión: optimizado para tablets

Android Versión 4.0 (Ice Cream Sandwich)

Lanzamiento: Cuarto trimestre de 2011, octubre o noviembre.

Funcionalidades: Diseñado para todas las plataformas (smartphones, tablets y netbooks de todas las medidas), pantalla principal con imágenes 3D, barras de estado y widgets redimensionables, reconocimiento facial, mejora de reconocimiento de voz; soporte USB para teclados y controles para PS3.

3.3.1.4 Arquitectura

Según Sanderson (2), la plataforma Android (Figura III - 3) está diseñada para ser más tolerante a fallos que muchos de sus predecesores.

Android usa Linux para los drivers de sus dispositivos, manejo de memoria, proceso de administración y la conectividad. Usualmente no se puede programar en esta capa directamente⁴².

El siguiente nivel contiene las librerías nativas de Android. Todo esto se escribe en C/C++ internamente, pero puede ser llamado alrededor de una interfaz java.

⁴² <http://code.google.com/android/what-is-android.html>

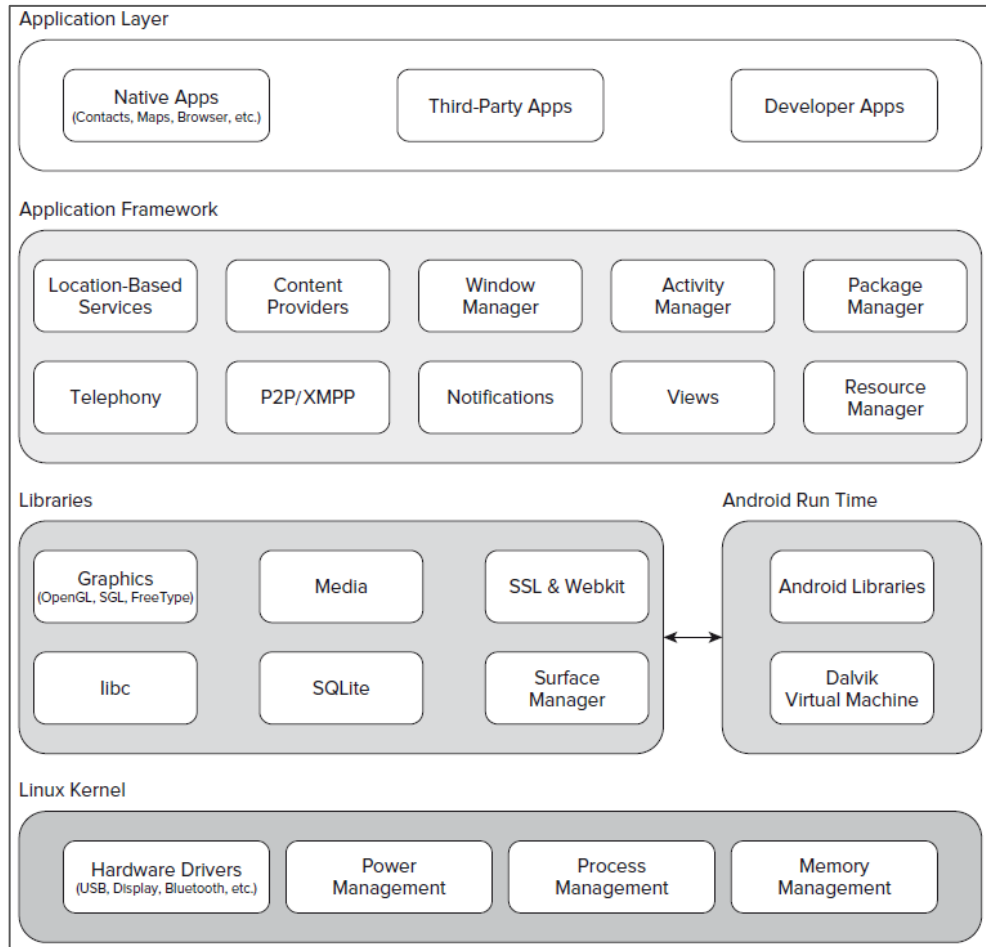


Figura III - 3 Arquitectura de Android

- **Aplicaciones:** Las aplicaciones base incluirán un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones están escritas en el lenguaje de programación Java.
- **Framework de aplicaciones:** Los desarrolladores tienen acceso completo a las APIs del framework usado por las aplicaciones base. La arquitectura está diseñada para simplificar el reuso de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. Este framework está formado por:

- Un extenso conjunto de Vistas tales como listas, cajas de texto, botones.
 - Content Providers que permiten a las aplicaciones acceder a información de otras aplicaciones o compartir su propia información.
 - Resource Manager, que proporciona acceso a recursos que no son código como pueden ser gráficos, cadenas de texto.
 - Notification Manager que permite a las aplicaciones mostrar alarmas personalizadas en la barra de estado.
 - Activity Manager, que gestiona el ciclo de vida de las aplicaciones.
-
- Librerías: Android incluye un set de librerías C/C++ usadas por varios componentes del sistema. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android, el cual interactúa con las librerías mediante JNI (Java Native Interface). Algunas son: System C library (implementación librería C estándar), librerías de medios, librerías de gráficos, 3d, SQLite, entre otras.
 - Runtime de Android: Android incluye un set 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 ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima.
 - Núcleo - Linux: Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila.

Anatomía de una aplicación de Android

Dentro de una aplicación de Android hay cuatro componentes principales: Activity, Listeners, Servicios y Content Provider. Todas las aplicaciones de Android están formadas por algunos de estos elementos o combinaciones de ellos.

Activity

Las Activities (o Actividades) son el elemento constituyente de Android más común. Para implementarlas se utiliza una clase por cada Actividad que extiende de la clase base Activity. Cada clase mostrará una interfaz de usuario, compuesta por Views (o Vistas). Cada vez que se cambie de Vista, se cambiará de Actividad, como por ejemplo en una aplicación de mensajería que se tiene una Vista que muestra la lista de contactos y otra Vista para escribir los mensajes. Cuando cambiamos de Vista, la anterior queda pausada y puesta dentro de una pila de historial para poder retornar en caso necesario. También se pueden eliminar las Vistas del historial en caso de que no se necesiten más. Para pasar de vista en vista, Android utiliza una clase especial llamada Intent.

Intent

Un Intent es un objeto mensaje y que, en general, describe que quiere hacer una aplicación. Las dos partes más importantes de un Intent son la acción que se quiere realizar y la información necesaria que se proporciona para poder realizarla, la cual se expresa en formato URI. Un ejemplo sería ver la información de contacto de una persona, la cual mediante un Intent con la acción ver y la URI que representa a esa persona se podría obtener. Relacionado con los Intents hay una clase llamada IntentFilter que es una descripción de que Intents puede una Actividad gestionar. Mediante los IntentFilters, el sistema puede resolver Intents, buscando cuales posee cada actividad y escogiendo aquel que mejor se ajuste a sus necesidades. El

proceso de resolver Intents se realiza en tiempo real, lo cual ofrece dos beneficios:

- Las actividades pueden reutilizar funcionalidades de otros componentes simplemente haciendo peticiones mediante un Intent.
- Las actividades pueden ser remplazadas por nuevas actividades con IntentFilters equivalentes.

Listeners

Los Listeners se utilizan para reaccionar a eventos externos (por ejemplo, una llamada). Los Listeners no tienen UI pero pueden utilizar el servicio Notification Manager para avisar al usuario. Para lanzar un aviso no hace falta que la aplicación se esté ejecutando, en caso necesario, Android la iniciará si se activa el Listeners por algún evento.

Servicios

Un Servicio es básicamente un código que se ejecuta durante largo tiempo y sin necesidad de UI, como puede ser un gestor de descarga en el cual se indican los contenidos a descargar y posteriormente el usuario puede acceder a una nueva Vista sin que el gestor se interrumpa. En caso de que haya múltiples servicios a la vez, se les puede indicar diferentes prioridades según las necesidades.

Content Provider

En Android, las aplicaciones pueden guardar su información en ficheros, BBDD SQLite. Pero en caso de que lo que se quiera sea compartir dicha información con otras aplicaciones, lo necesario es un Content Provider. Un Content Provider es una clase que implementa un conjunto estándar de métodos que permite a otras aplicaciones guardar y obtener la información que maneja dicho Content Provider.

Android Manifest

En Android existe un archivo XML llamado AndroidManifest que, aunque no forme parte del código principal de la aplicación, es necesario para su correcto funcionamiento. Este archivo es el fichero de control que le dice al sistema que tiene que hacer con todos los componentes anteriormente mencionados en este apartado que pertenecen a una aplicación en concreto.

Tipos de aplicación de Android

Un Android Package (.apk) es el fichero que contiene el código de la aplicación y sus recursos, y que posteriormente se instala en el dispositivo para poder ejecutar la aplicación.

Tareas

Una tarea en Android es lo que el usuario ve como una aplicación y el desarrollador ve como una o más Actividades donde el usuario interacciona y va pasando de Vista en Vista. Dentro de las tareas, una actividad toma el papel de punto de entrada (será la primera en mostrarse cuando se ejecute la aplicación) y las demás, si hay, formarán parte de la misma tarea, a la espera de ser instanciadas.

Procesos

En Android, los procesos se ejecutan a nivel de kernel y el usuario normalmente no tiene constancia de ellos. Todo el código de la aplicación se suele ejecutar en un proceso dedicado pero también se puede especificar si sólo se quiere que se ejecute en el proceso una determinada clase o componente de la aplicación. Los principales usos de los procesos son:

- Mejorar la estabilidad o seguridad de las aplicaciones.

- Reducir la sobrecarga de proceso ejecutando el código de múltiples aplicaciones en el mismo proceso.
- Ayudar al sistema a gestionar los recursos separando partes de código pesado en un proceso separado que puede ser eliminado independientemente de otras partes de la aplicación.

Threads

En lo referente a threads, Android evita la creación de threads adicionales por parte de un proceso, manteniendo la aplicación en un sólo thread al menos que no los cree la propia aplicación. Esto repercute de manera importante en las llamadas a instancias a Actividades, Listeners y Servicios, ya que sólo pueden ser hechas por el thread principal del proceso en el que están corriendo. Por otro lado, al no crearse un thread por cada instancia, dichas instancias no deben realizar operaciones largas o bloqueantes cuando son llamadas, de lo contrario, bloquearían todos los demás componentes del proceso.

Ciclo de vida de una aplicación de Android

Cada aplicación de Android corre en su propio proceso, el cual es creado por la aplicación cuando se ejecuta y permanece hasta que la aplicación deja de trabajar o el sistema necesita memoria para otras aplicaciones. Una característica fundamental de Android es que el ciclo de vida de una aplicación no está controlado por la misma aplicación sino que lo determina el sistema a partir de una combinación de estados como pueden ser que aplicaciones están funcionando, que prioridad tienen para el usuario y cuanta memoria queda disponible en el sistema. De esta manera, Android sitúa cada proceso en una jerarquía de importancia basada en los estados comentados, como se puede ver a continuación.

1. Un proceso en primer plano es uno que se requiere para lo que el usuario está actualmente haciendo. Se considera en primer plano si:

- Está ejecutándose una Actividad perteneciente a la pantalla con la que el usuario está interactuando.
 - Está ejecutando un BroadcastReceiver.
 - Está ejecutándose un servicio.
2. Un proceso visible es aquel que contiene una Actividad que es visible al usuario mediante la pantalla pero no en primer plano (esta pausada). Este proceso solo se eliminará en caso de que sea necesario para mantener ejecutándose los procesos en primer plano.
 3. Un proceso de servicio es aquel que contiene un servicio que ha sido inicializado. No son directamente visibles al usuario y el sistema los mantendrá a no ser que no pueda servir los dos anteriores.
 4. Un proceso en background es aquel que acoge una actividad que no está actualmente visible al usuario. Mientras que dichos procesos implementen bien su propio ciclo de vida, el sistema puede eliminarlos para dar memoria a cualquiera de los 3 servicios anteriores.
 5. Un proceso vacío es aquel que no contiene ningún componente activo de ninguna aplicación. La única razón para mantener dicho proceso es para mejorar sus inicializaciones posteriores a modo de caché.

Para comprender mejor el ciclo de vida de una aplicación de Android, en la (Figura III - 4) se muestra el diagrama de flujo de dicho ciclo, mencionando también los métodos que se llaman durante el transcurso del mismo.

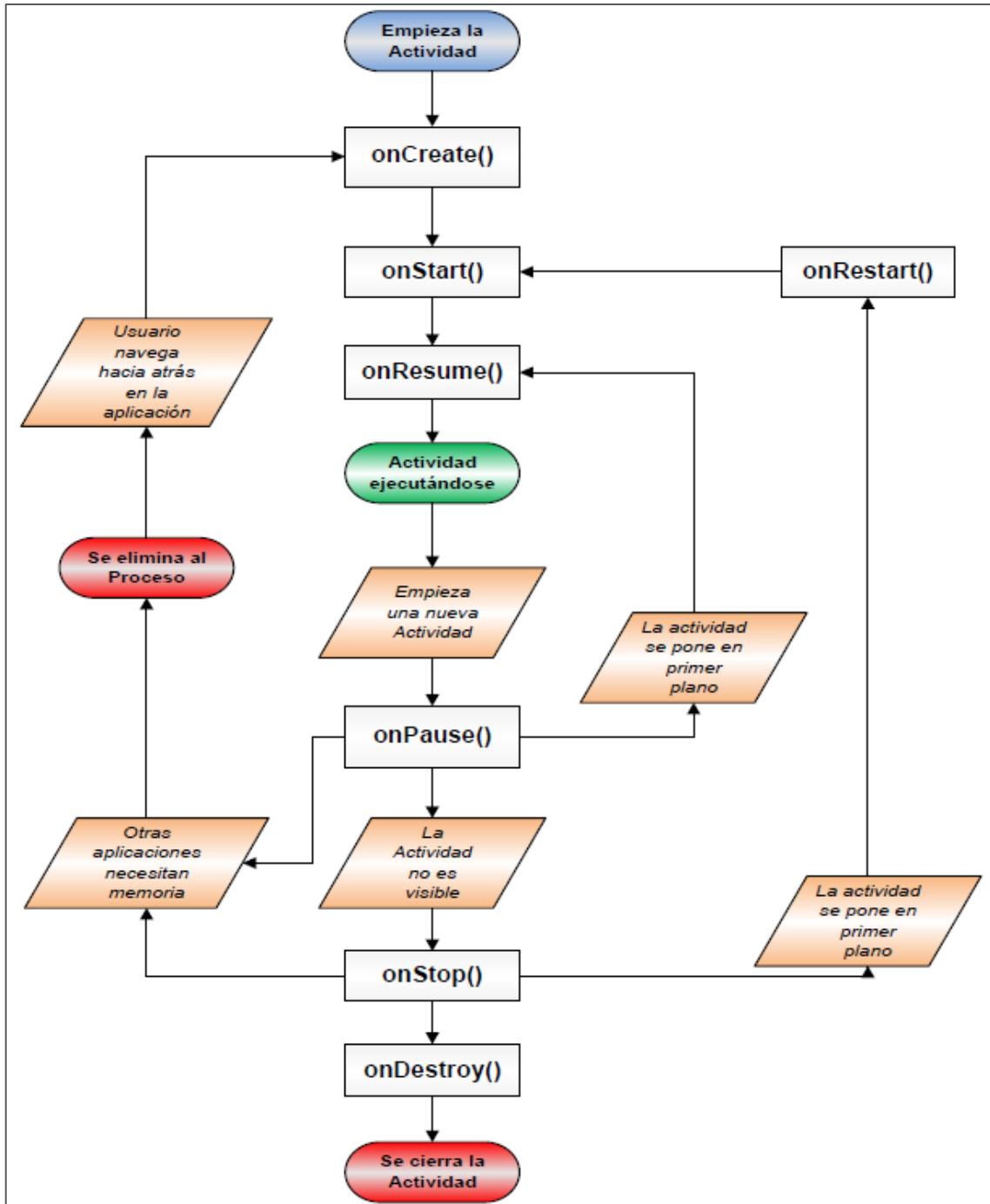


Figura III - 4 Ciclo de vida de una aplicación en Android

Recursos

Los recursos en Android son ficheros externos que no son código y que son usados y compilados dentro de la aplicación. Android soporta diferentes tipos de recursos, incluido XML, PNG y JPEG. Los ficheros XML tienen diferente formato dependiendo de que describan.

Los recursos se exteriorizan respecto al código. Los ficheros XML se compilan dentro un archivo binario por razones de eficiencia y los strings se comprimen en una forma más eficiente de almacenamiento. Es por esta razón que hay diferentes tipos de recursos en Android. En general, hay recursos de tres tipos: ficheros XML, bitmaps y ficheros raw (por ejemplo, ficheros de sonido). En la TABLA III - I se muestran los tipos de recursos posibles que hay en Android.

TABLA III - I Tipos de Recursos en Android

Tipo	Descripción
Anim	Ficheros XML que son compilados en objetos de animaciones.
Drawable	Ficheros con formato .png, .png, .jpg que se compilan en un fichero bitmap.
Layout	Ficheros XML que se compilan en vistas de pantalla.
Values	Ficheros XML que se compilan en varios tipos de recursos, para definir, por ejemplo, arrays, colores, strings.
XML	Ficheros XML arbitrarios que son compilados y pueden ser llamados en tiempo de ejecución.
Raw	Ficheros arbitrarios que son copiados directamente al dispositivo.

Fuente: Reto Meier, Android Application Development

3.3.1.5 SDK (Kit de Desarrollo de Software)

Según Sanderson (2), el kit de desarrollo de software de android (software development kit o SDK) incluye un conjunto de herramientas de desarrollo,

tales como un debugger, librerías, un emulador, documentación, código de ejemplo y tutoriales. Está soportado en Sistemas Operativos Windows, Linux y Mac.

El entorno de desarrollo (integrated development environment o IDE) oficialmente soportado es Eclipse conjuntamente con el plugin ADT proporcionado por Google.

Desde noviembre del 2007 a septiembre del 2008 han ido surgiendo nuevas actualizaciones del SDK siendo la última la 18, la cual soluciona algunos errores de las anteriores y añade nuevas funcionalidades y cambios en las APIs.

En cuanto a las herramientas del SDK se tiene:

- **Android Eclipse Views.-** El plugin de Android para Eclipse añade un número de vistas para poder debugar la aplicación. Una de las más importantes es el LogCat, un visor de logs del dispositivo que puede ser filtrado en 5 categorías:

V — Verbose (Prioridad más baja)

D — Debug

I — Info

W — Warning

E — Error

F — Fatal

S — Silent (Alta prioridad)

- **Emulador Console.-** Algunas funciones en tiempo real del emulador, como el envío de SMS, pueden ser accedidas usando la consola del emulador mediante telnet a localhost y puerto, normalmente, 5554.

- **Android Debug Bridge (ADB).**- Proporciona acceso a funciones del emulador, incluido el redireccionamiento de puertos.
- **Dalvik Debug Monitor Service (ddms).**- La "Dalvik Debug Monitor Service" es un herramienta integrada con la "Dalvik Virtual Machine", y nos permite administrar los procesos que corren en una instancia de emulador/dispositivo y nos asiste en la depuración de ellos. Podemos usar esta herramienta para terminar la ejecución de un proceso, seleccionar un determinado proceso para depurar, generar reportes a partir de información de bitácoras, examinar el "heap" y la información de "thread, tomar fotografías de la pantalla del emulador/dispositivo y mucho más.
- **Android Debug Bridge (adb).**- La herramienta "adb" nos permite instalar nuestras aplicaciones (archivos ".apk") en una instancia de emulador/dispositivo y acceder una instancia de emulador/dispositivo usando comandos de línea. También podemos utilizarlo para enlazar un depurador estándar al código de una aplicación que esté corriendo en una instancia de emulador/dispositivo.
- **Android Asset Packaging Tool (aapt).**- La herramienta "aapt" te permite crear archivos ".apk", los cuales contienen las imágenes binarias de tu código y recursos de tus aplicaciones.
- **Android Interface Description Language (aidl).**- Te permite generar código para una interface de interproceso, tal como la que un servicio podría utilizar.
- **sqlite3.**- Esta herramienta ha sido incluida para comodidad de los desarrolladores. Nos provee acceso a los archivos de datos "SQLite" creados y usados por las aplicaciones Android.

- **Mksdcard.-** Esta herramienta nos ayuda a crear una imagen de disco que podemos usar con el emulador, para simular la presencia de una tarjeta de almacenamiento externa (tal como una tarjeta "SD").
- **Dx.-** La herramienta "dx" convierte los archivos de "bytecode"⁴³ estándar (".class") en archivos "Android bytecode" (".dex").

3.3.1.6 Sistemas Operativos

El SDK se encuentra disponible para los siguientes sistemas operativos:

- Linux (Todas las Distribuciones)
- Windows
- Mac OS X

3.3.1.7 Lenguajes de Programación

Para trabajar con Android el lenguaje de programación más óptimo es Java, ya que la mayoría de programas para Android están escritos en este lenguaje y utilizan las API de la biblioteca Java 5 Standard Edition (SE).

Android también utiliza un compilador estándar Java para compilar el código fuente, lo que lo hace que Java sea un lenguaje completamente integrado en Android.

Opcionalmente se puede desarrollar en C o C++ utilizando un NDK (Kit de Desarrollo Nativo) para incrementar velocidad en el funcionamiento de las aplicaciones ejemplo en el desarrollo de juegos.

⁴³ <http://es.wikipedia.org/wiki/Bytecode>

3.3.1.8 Entornos de Programación

Después de decidir el sistema operativo y el lenguaje de programación de la aplicación, necesitamos una herramienta para poder programar en este lenguaje.

Las IDE más populares para Android son Eclipse y NetBeans. Ambas son gratuitas y de código abierto. Para escoger uno nos basamos en la amplia información recopilada que determina la compatibilidad entre el entorno de desarrollo y el emulador o el terminal sin tener que realizar pasos intermedios.

Android Development Tools Plugin para Eclipse IDE

El "ADT plugin" agrega poderosas extensiones al ambiente integrado de Eclipse haciendo que la creación y depuración de tus aplicaciones Android sea fácil y rápida. Si tú desarrollas estás usando Eclipse, el "ADT plugin" te da un increíble estímulo para el desarrollo de las aplicaciones Android. A continuación alguna de sus características:

- Provee acceso a otras herramientas de desarrollo de Android desde el entorno de Eclipse IDE. Por ejemplo, "ADT" te permite el acceso a muchas de las capacidades de la herramienta "DDMS", tales como tomar fotografías de la pantalla, administrar el redireccionamiento de puertos, fijar puntos de quiebre y examinar la información "thread" y procesos directamente dentro de Eclipse.
- Provee un asistente para la creación de proyectos Android, el cual te ayuda a crear rápidamente todos los directorios y archivos necesarios para crear una nueva aplicación Android.

- Automatiza y simplifica el proceso de construcción de una aplicación.
- Provee un editor de código Android que te ayuda a escribir XML válido para el archivo "AndroidManifest.xml".

3.3.1.9 Distribución de Aplicaciones

La publicación de las aplicaciones se las realiza de las siguientes formas:

- Instalando el ejecutable (.apk) en cada uno de los terminales si previo registro.
- Publicando en Google Play

Pasos para la distribución de aplicaciones

1. Para subir una aplicación lo primero es probar que funcione, luego se tiene que firmar la aplicación⁴⁴.
2. Registrarse en Google Play cómo desarrollador. Para esto hay que ingresar a Google Play para desarrolladores, ingresar una cuenta de correo de gmail y dar algunos datos.
3. Crear una cuenta en Google Checkout y abonar la tarifa de 25 \$ para toda la vida.
4. Una vez pagada la cuota, podemos subir la aplicación (.apk), varias fotos del logo de nuestra aplicación, más algunas imágenes de cómo es la aplicación y finalmente dar una descripción de la aplicación en los idiomas que deseamos.

3.3.1.10 Tienda de aplicaciones

Google Play

⁴⁴ <http://ungranmundolibre.blogspot.com/2011/08/firmando-tu-aplicacion.html>

Google Play (antes llamado Android Market) es una tienda de software en línea desarrollada por Google para los dispositivos Android. Es una aplicación ("app") que está preinstalada en la mayoría de los dispositivos Android y que permite a los usuarios buscar y descargar aplicaciones publicadas por desarrolladores terceros, alojada en Google Play. Los usuarios también pueden buscar y obtener información sobre aplicaciones en esta página web. En enero de 2012, Google Play disponía de más de 500.000 aplicaciones.

Su archivo ejecutable se llama Vending.apk

El 6 de marzo de 2012, Android Market fue rebautizado como Google Play.

3.3.1.11 Instalación de Eclipse con el SDK de Android

A continuación se describen los pasos necesarios para empezar a desarrollar y a entender aplicaciones para Android. Las instrucciones de instalación fueron realizadas en el entorno de desarrollo Eclipse versión 3.7.2, conocida como Indigo. El SDK de Android puede correr en los sistemas operativos como Mac OS X, linux o Windows.

Descargar el SDK de Android

Android es una plataforma de software libre, por lo que cuenta con un SDK disponible para todo desarrollador que lo desee que incluye, entre otros elementos, el conjunto completo de API que este sistema soporta. Para descargarlo, basta con visitar la web de Android y asegurarse de acceder a la última versión publicada (durante la redacción de estas líneas, la última versión es la 4.0).

Una vez descargado el SDK, es necesario descomprimirlo. La ubicación de los ficheros resultantes no es relevante, pero conviene recordar la ruta para pasos posteriores.

Descargar Eclipse Indigo

La descarga de Eclipse no es muy diferente al SDK de Android. La web de Eclipse ofrece multitud de versiones de este entorno de desarrollo según las necesidades del desarrollador. En este caso, es suficiente con obtener la versión 3.7.2, denominada Indigo.

Finalizada la descarga, no se realiza ningún proceso de instalación; simplemente se debe descomprimir los ficheros y pulsar el ejecutable para abrir la aplicación. La primera vez que se inicie Eclipse, pide al usuario una localización para el workspace, donde se ubicarán por defecto todos los proyectos desarrollados.

Instalación el plug-in de Android

El siguiente paso consisten en instalar un plug-in específico de Android para la plataforma Eclipse. Esta herramienta, llamada ADT (Android Development Tools), facilita enormemente la creación de proyectos, su implementación, depuración y ejecución, por lo que es altamente recomendable si se quiere trabajar con Android.

Para instalar el plug-in ADT en Eclipse Indigo, es necesario seguir las siguientes indicaciones:

1. Iniciar Eclipse
2. Seleccionar la pestaña Help > Software Updates. Esta acción abrirá una nueva ventana llamada Software Updates and Add-ons.
3. Pinchar en la pestaña Available Software y pulsar el botón Add Site.
4. Introducir la siguiente URL y pulsar OK:

<https://dl-ssl.google.com/android/eclipse/>

5. Volviendo a la ventana Software Updates and Add-ons, marcar la casilla correspondiente a Developer Tools y pulsar el botón Install. Se abrirá una nueva ventana.
6. Cerciorarse de que las opciones Android Developer Tools y Android Editors están marcadas y pulsar el botón Finish.

El proceso de instalación dará comienzo y puede llevar algunos minutos. Con el fin de que los cambios tengan efecto, es necesario reiniciar Eclipse.

Referenciar el SDK de Android

Tras abrir de nuevo Eclipse, debe indicarse en las preferencias de Eclipse la localización del SDK a utilizar para los proyectos de Android:

1. Seleccionar la pestaña Window > Preferences, lo que abrirá una nueva ventana.
2. Elegir Android en el panel izquierdo.
3. Pulsar el botón Browse e indicar la ruta del SDK de Android.
4. Pulsar el botón Apply y después OK.

Instalación de Componentes del SDK

1. Abrir el administrador del SDK de Android.
2. Seleccionar la versión del API y las herramientas Android 4.0 (API 14)

3.3.2 iOS

Después de tener en cuenta los relevantes de cada uno de los sistemas operativos así como el estado actual del sistema operativo, iOS es una tecnología que ofrece grandes prestaciones a los desarrolladores, proporcionando una amplia documentación oficial y no oficial.

iOS es un sistema operativo desarrollado por Apple Inc. para los dispositivos móviles iPod touch, iPhone e iPad. . Está basado en una variante del Mach kernel que se encuentra en Mac OS X. El iOS incluye el componente de software “Core Animation” de Mac OS X v10.5 que, junto con el PowerVR MBX el hardware de 3D, es responsable de las animaciones usadas en el interfaz de usuario.

iOS tiene 4 capas de abstracción:

- la capa del núcleo del sistema operativo,
- la capa de Servicios Principales,
- la capa de Medios de comunicación y
- la capa de Cocoa Touch.

El sistema operativo ocupa bastante menos de medio gigabyte del total del dispositivo, de 8 GB o de 16 GB. Esto se realizó para poder soportar futuras aplicaciones de Apple.

3.3.2.1 Características

Interfaz de usuario

La interfaz de usuario de iOS se basa en con el concepto de manipulación mediante gestos multitáctil. Los elementos de la interfaz se componen por deslizadores, interruptores y botones. La respuesta es inmediata y se provee de una interfaz fluida. La interacción con el sistema operativo se realiza mediante gestos como deslizar, tocar y pellizcar. Acelerómetros y Giroscopios internos son utilizados por algunas aplicaciones para responder a movimientos y gestos, como sacudir el aparato (en campos de texto es usado para deshacer y rehacer) o rotarlo (se suele usar para cambiar de posición vertical a modo paisaje).

Pantalla principal

La pantalla principal (llamada «SpringBoard») es donde se ubican los iconos de las aplicaciones y el Dock en la parte inferior de la pantalla donde se pueden

anclar aplicaciones de uso frecuente, aparece al desbloquear el dispositivo o presionar el botón de inicio. La pantalla tiene una barra de estado en la parte superior para mostrar datos, tales como la hora, el nivel de batería, y la intensidad de la señal.

Carpetas

Con iOS 4 se introdujo un sistema simple de carpetas en el sistema. Se puede mover una aplicación sobre otra y se creará una carpeta, y así se pueden agregar más aplicaciones a esta mediante el mismo procedimiento. Pueden entrar hasta 12 y 20 aplicaciones en el iPhone y iPad respectivamente. El título de la carpeta es seleccionado automáticamente por el tipo de aplicaciones dentro de ella, pero puede ser editado por el usuario.

Centro de notificaciones

Con la actualización iOS 5, el sistema de notificaciones se rediseñó por completo. Las notificaciones ahora se colocan en un área la cual se puede acceder mediante un desliz desde la barra de estado hacia abajo. Al hacer un toque en una notificación, el sistema abre la aplicación que envió la notificación.

Aplicaciones

La pantalla inicial de iOS contiene varias aplicaciones, algunas de las cuales están ocultas por defecto y pueden ser activadas por el usuario mediante la aplicación "Configuración". La tienda de Apple aloja una multitud de aplicaciones en varias categorías para iOS.

Multitarea

Antes de iOS 4, la multitarea estaba reservada para aplicaciones por defecto del sistema. A Apple le preocupaba los problemas de batería y rendimiento si se permitiese correr varias aplicaciones de terceros al mismo tiempo. A partir de iOS 4, dispositivos de tercera generación y posteriores soportan el uso de 7 APIs para multitarea, específicamente:

- Audio en segundo plano
- Voz IP
- Localización en segundo plano
- Notificaciones push
- Notificaciones locales
- Completado de tareas
- Cambio rápido de aplicaciones

Sin embargo, no consiste en una verdadera multitarea, pues las aplicaciones ajenas al SO, quedan congeladas en segundo plano no recibiendo un solo ciclo de reloj del procesador.

Game Center

Fue anunciado en el evento donde se presentó iOS 4 el 8 de Abril, 2010. Game Center se lanzó en junio de 2010 para los iPhone y iPods Touch con iOS 4 (excepto para el iPhone 2G y iPod Touch 1g). En iOS 5 se perfeccionó, pudiendo agregar una foto a tu perfil, pudiendo ver los amigos de tus amigos y pudiendo encontrar adversarios con recomendaciones de nuevos amigos en función de tus juegos y jugadores favoritos.

Tecnologías no admitidas

iOS no permite Adobe Flash⁴⁵ ni Java. Steve Jobs escribió una carta abierta donde critica a Flash por ser inseguro, con errores, consumir mucha batería, ser incompatible con interfaces multitouch e interferir con el servicio App Store.¹¹ En cambio iOS usa HTML5⁴⁶ como una alternativa a Flash. Esta ha sido una característica muy criticada tanto en su momento como la actualidad. Sin embargo por métodos extraoficiales se le puede implementar aunque conllevaría la pérdida de la garantía.

⁴⁵ http://es.wikipedia.org/wiki/Adobe_Flash_Player

⁴⁶ <http://es.wikipedia.org/wiki/HTML5>

Durante los últimos meses de 2010 en el Silverlight Firestarter Event se especulaba sobre la inclusión en iOS de Silverlight, y con "transmuxing" se revelaba un gran avance para iOS en esta dirección, pero con los últimos sucesos en los que Microsoft considera redirigir sus intereses a HTML5 el futuro de Silverlight es incierto y por el momento es casi imposible encontrar contenido Silverlight disponible en dispositivos con iOS

3.3.2.2 Restricciones

El SDK se puede descargar gratis, pero para publicar el software es necesario registrarse en el Programa de Desarrollo del iOS, un paso que requiere el pago y la aprobación por parte de Apple. Durante el proceso, se entregan al desarrollador unas claves firmadas que permiten subir una aplicación a la tienda de aplicaciones de Apple.

Las aplicaciones pueden ser distribuidas de 3 formas: a través de la App Store de Apple, por parte de una empresa a sus empleados, o sobre una red "Ad-hoc" de hasta 100 iPhones.

Los desarrolladores son libres de poner cualquier precio para que sus aplicaciones sean distribuidas por la tienda de Apple. A cambio de ello la compañía se queda con un porcentaje de los ingresos (30%). Los desarrolladores pueden optar a poner estas aplicaciones de uso gratis, y no pagar ningún gasto para distribuir el programa más allá de la cuota de socio.

Este modelo de distribución para el software iOS hace imposible liberar el software basado en el código autorizado con GPLv3 (Licencia Pública General de GNU). Cualquier código que modifique el código autorizado bajo GPLv3 también debe tener licencia como GPLv3. Un desarrollador no es capaz de distribuir una aplicación autorizada bajo el GPLv3 sin distribuir también las claves para firmar (que Apple posee) para permitir la carga de las versiones modificadas de aquel software.

3.3.2.3 Versiones de iOS

En el momento de la escritura, el IOS (iPhone OS) está en su cuarta revisión es decir, la versión 4.0. Sus principales versiones son las siguientes:

- **1.0** Versión inicial del iPhone
- **1.1** Las características adicionales y equis error fi para 1.0
- **2.0** Fecha de lanzamiento con el iPhone 3G, viene con la App Store.
- **2.1** Las características adicionales y corrección de error inalámbrica de 2.0
- **2.2** Las características adicionales y equis error inalámbrica de 2,1
- **3.0** Gran Lanzamiento Tercera del iPhone OS
- **3.1** Las características adicionales y equis error música para la versión 3.0
- **3.2** Este comunicado es la versión de la única IPAD, consulte la barra lateral de lo que es nuevo en el iPhone OS 3.2.
- **4.0** versión Cuarta principales del iPhone OS. Renombrado como IOS. Esta versión está diseñada para el nuevo iPhone 4 y también es compatible con los dispositivos más antiguos, como el iPod touch e iPhone⁴⁷.

3.3.2.4 Arquitectura de iOS

La Figura III - 5, muestra las capas de abstracción que componen el Mac OS X y el iPhone OS (que es utilizado por el iPhone, iPod touch, y IPAD).

⁴⁷

<http://developer.apple.com/iphone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSFrameworks/iPhoneOSFrameworks.html>



Figura III - 5 Arquitectura de Mac OS X y iPhone OS

La capa inferior es el sistema operativo subyacente, que es el base del sistema operativo. Tiene a su cargo de gestión de memoria, el sistema de archivos, creación de redes, y otras tareas del sistema operativo, y que interactúa directamente con el hardware. La capa núcleo operativo consiste en componentes como los siguientes:

- Kernel OS X
- Mach 3.0
- BSD
- Sockets
- Seguridad
- Administración de energía
- Llaverio
- Certificados
- sistema de archivos
- Bonjour

La capa Core Services proporciona una abstracción sobre los servicios prestados en la capa de núcleo del sistema operativo. Es proporciona acceso fundamental a los servicios de iPhone OS y se compone de los siguientes componentes:

- Colecciones
- Libreta de direcciones
- Redes
- Acceso a los archivos
- SQLite
- Básicas Ubicación
- Red de Servicios Básicos
- Threading
- Preferencias
- Utilidades URL

La capa de los medios de comunicación proporciona los servicios multimedia que se pueden utilizar en tu iPhone y las aplicaciones de IPAD.

Se compone de los siguientes componentes:

- Core Audio
- OpenGL
- Mezcla de audio
- Grabación de audio
- Reproducción de vídeo
- JPG, PNG, TIFF
- PDF
- Quartz
- Animación Básicos
- OpenGL ES

La capa de Cocoa Touch proporciona una capa de abstracción para exponer las diferentes bibliotecas de la programación para el iPhone y el iPad, tales como las siguientes⁴⁸:

- Multi-Touch de eventos
- Controles Multi-Touch
- Acelerómetro
- Ver Jerarquía
- Localización
- Alertas
- Vistas Web
- Selector de personas
- Selector de Imagen
- Controladores

En la programación del iOS, todas las funcionalidades en cada capa se exponen a través de distintos marcos que utilizará en el proyecto.

3.3.2.5 SDK de iOS

Según Wei Meng Lee [4], iOS comparte base con Mac OS X, la cadena de instrumentos para desarrollar aplicaciones para iOS (Figura III - 6) está también basada en Xcode e incluye compiladores cruzados para el procesador ARM y un emulador de iOS llamado Aspen. El lenguaje de programación principal para iOS, al igual que en Mac OS, es Objective-C.

⁴⁸ El iOS es una arquitectura muy similar a la de Mac OS X, excepto que la capa superior es Cocoa Touch para el iOS en lugar del marco del Cacao.

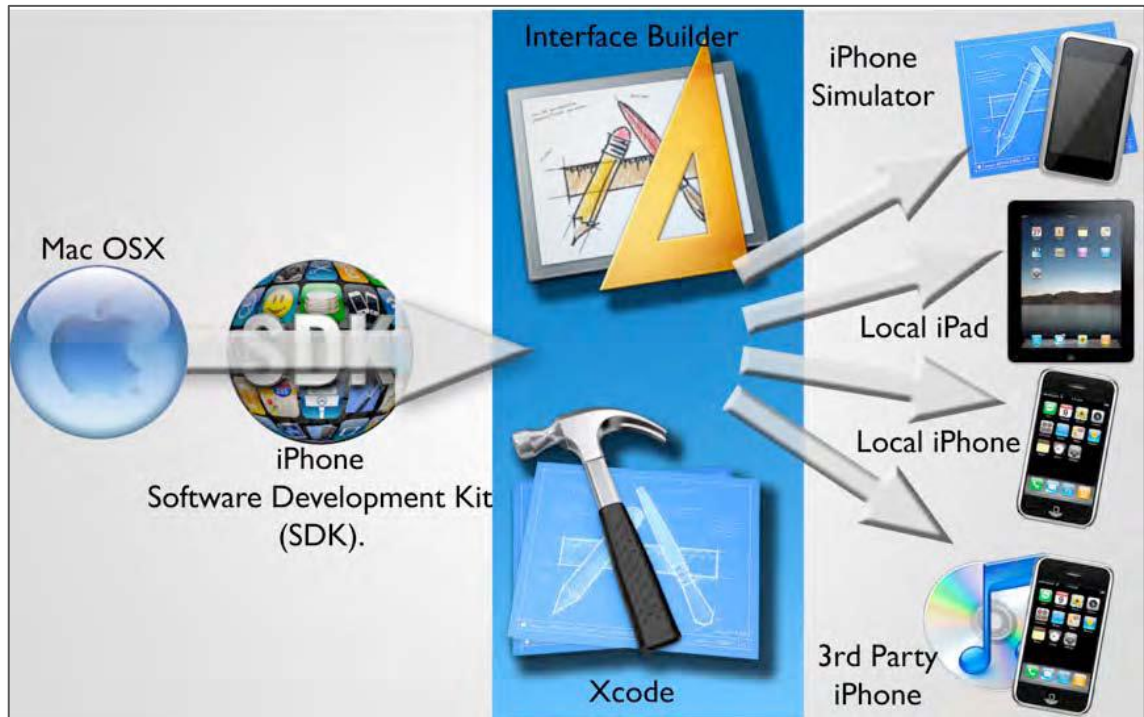


Figura III - 6 Esquema del SDK de iOS

Como iOS utiliza una variante del mismo núcleo de XNU que se encuentra en Mac OS X, la cadena de herramientas utilizada para el desarrollo en iOS también se basa en Xcode.

El SDK se desglosa en los siguientes grupos:

- Cocoa Touch
 - Multi-touch eventos y controles
 - Acelerómetro apoyo
 - Localización (i18n)
 - Cámara de soporte
- Medios de comunicación
 - OpenAL
 - mezcla de audio y grabación
 - La reproducción de vídeo
 - Formatos de archivo de imagen
 - Cuarzo

- Core Animation
- OpenGL ES
- Core Services
 - Redes
 - Embedded SQLite base de datos
 - Core Location
 - Temas
 - CoreMotion
- Mac OS X Kernel
 - TCP / IP
 - Sockets
 - Gestión de la energía
 - El sistema de archivos
 - Seguridad

Junto con la cadena de herramientas de Xcode, el SDK contiene el iPhone e iPad Simulador, unos programas que sirve para emular la apariencia del iPhone e iPad en el escritorio del desarrollador. Originalmente llamado el simulador Aspen, fue renombrado con la versión Beta 2 del SDK.

3.3.2.6 Sistemas operativos

El SDK requiere una Mac con procesador Intel corriendo Mac OS X Snow Leopard o posterior. Otros sistemas operativos, incluyendo Microsoft Windows y las versiones anteriores de Mac OS X, no son compatibles.

3.3.2.7 Lenguajes de Programación

El lenguaje nativo para desarrollar aplicación para iOS es Objective-C. Objective-C es un lenguaje de programación orientado a objetos creado como un superconjunto de C pero que implementase un modelo de objetos parecido al de Smalltalk.

3.3.2.8 Entorno de Programación

Las herramientas que forman parte del entorno de desarrollo de la plataforma:

Xcode: Xcode es la pieza esencial del entorno. Es un IDE muy completo, en el que podemos editar código fuente, acceder a un completa volumen de documentación, y hasta un debugger gráfico. Xcode está construido por numerosos módulos Open Source (como el compilador gcc y el debugger gdb).

Instruments: Instruments sirve para monitorear la aplicación en desarrollo y sintonizar finamente su performance, en un maravilloso entorno gráfico. Instruments está basado en DTrace, una herramienta Open Source desarrollada por Sun Microsystems. Es esencial a la hora de identificar pérdidas de memoria y otros bugs difíciles de rastrear.

Dashcode: Dashcode fue diseñada para desarrollar widgets para el Dashboard de Mac OS X. La versión del SDK de iOS es básicamente la misma que la de Mac OS X, y fue incluida para facilitar el desarrollo de aplicaciones Web para el iOS.

Simulator: el simulador de iOS permite probar las funcionalidades básicas de la aplicación en desarrollo. El simulador corre un sistema operativo parecido al real. Cuando se trabaja con el simulador, Xcode compila para x86, en lugar de ARM (que es la arquitectura del sistema real). Para compilar código ARM es necesario firmar la aplicación, lo que requiere la certificación de Apple.

Interface Builder: Su uso elemental es el desarrollo de la Interfaz Gráfica de Usuario (GUI), sin embargo es mucho más que eso. Interface Builder es donde los diferentes módulos se conectan, y es también donde se le da vida a muchos de los módulos precompilados que se verán en la pantalla.

3.3.2.9 Distribución de Aplicaciones

Para que el SDK del iOS compile código ARM que es la arquitectura del sistema real es necesario firmar la aplicación, lo que requiere la certificación de Apple. Una nueva firma es necesaria para distribuir la aplicación y por supuesto, si queremos que esta esté disponible en el appStore (página oficial de distribución de aplicaciones para el entorno Mac OS) deberemos solicitar la aprobación de Apple.

El proceso completo para desarrollar, compilar y publicar una aplicación para iOS es el siguiente:

1. Inscribirse en el del iOS Developer Program⁴⁹., aceptar todas las condiciones legales, y descargar el SDK. Haciendo esto podemos empezar a escribir nuestras aplicaciones, compilarlas y probarlas en el simulador. Pero todavía no podremos instalarlas en nuestro dispositivo.
2. Inscribirse en el programa para desarrolladores. Después de completar todos los formularios y leer más legales, podemos enviar la solicitud y esperar la respuesta hasta el siguiente día laboral. Esto tiene un costo de U\$99, y con él obtenemos el derecho a ejecutar nuestras aplicaciones en nuestro propio dispositivo(iphone, ipad).
3. Certificados. Por defecto, un iPhone tan solo puede ejecutar aplicaciones firmadas por Apple y como no es posible estar enviándoles el código cada vez que se quiere probar algo, es necesario crear un perfil con el identificador de nuestro dispositivo (accesible mediante las Xcode) e introducirlo en un formulario de la web de Apple. Luego para generar el certificado con el que firmar nuestras aplicaciones vamos a "Keychain Access" y en "Asistente para Certificados" generaremos uno nuevo. Si todo ha ido bien ya podremos experimentar nuestras aplicaciones en iOS.

⁴⁹ <https://developer.apple.com/programs/ios/>

4. Hasta no hace mucho, se prohibía a los desarrolladores hablar sobre el SDK, lo que dificultaba enormemente el desarrollo en sí. Trabajar con cualquier plataforma nueva conlleva dificultades que habitualmente se resuelven en foros y grupos para desarrolladores. Por suerte, Apple ha abandonado esta política tan cerrada y ahora se puede intercambiar códigos, conocimientos, etc.
5. Para distribuir es necesario otro certificado especial de distribución que hay que solicitar siguiendo el mismo proceso de antes. Acceder a Keychain Access, solicitar, aprobar, descargar e instalar. Al igual que en el punto 3, también necesitamos hacer funcionar un nuevo perfil de distribución en Xcode; proceso sobre el que Apple facilita muchas páginas de información con toda clase de capturas en las que se detalla el modo de reconfigurar el proyecto de Xcode para que utilice este certificado de distribución.
6. Enviar la aplicación a Apple a través de un nuevo formulario web con información sobre esta (nombre, descripción, versión...), su icono y algunas capturas de pantalla.
7. Una vez enviada, la aplicación pasa a la cola de revisión, sin que sepamos en ningún momento lo que se está haciendo, el estado de la solicitud, o cuanto queda para que termine el proceso. Si la aplicación no es aceptada por el motivo que sea, no habrá ayuda al respecto, solo se puede hacer los arreglos necesarios y volverla a enviar a través de un enlace con el que enviamos nuevas versiones de una misma aplicación. Si los problemas continúan, habrá que esperar una semana para saberlo y ver si la solución que has dado resuelve satisfactoriamente el problema.

8. Si la aplicación ha sido aceptada, ahora queda esperar que dé frutos la aplicación frente de los 10 millones de clientes potenciales que la podrán comprar.

3.3.2.10 Tienda de aplicaciones App Store

App Store es un servicio para el iPhone, el iPod Touch, el iPad, Mac OS X Snow Leopard y Mac OS X Lion, creado por Apple Inc., que permite a los usuarios buscar y descargar aplicaciones informáticas de iTunes Store o Mac App Store en el caso de Mac OSX, desarrolladas con el iOS SDK y publicadas por Apple. Estas aplicaciones están disponibles para ser compradas o bien gratuitas, dependiendo de cada una. Las aplicaciones pueden ser descargadas directamente al iPhone o al iPod Touch por medio de una aplicación del mismo nombre, aunque App Store también está disponible en el interior del programa informático iTunes.

Si bien Apple ha manifestado que no espera obtener ganancias de la tienda, Piper Jaffray predijo que App Store podía crear un mercado rentable con ingresos que excedan los mil millones de dólares anuales para la compañía. Apple otorga el 70% de los ingresos de la tienda directamente al vendedor de la aplicación y el 30% corresponde a Apple.

Apple revisa la aplicación y si la aprueba se publica en en el App Store, se requiere pagar una tasa de \$99 dólares para subir aplicaciones a la tienda

3.3.2.11 Instalación de Xcode en Mac OS X Snow Leopard



Figura III - 7 Xcode

Para iniciar debemos descargar Xcode a través de la aplicación App Store, que se encuentra en nuestra Mac. Abrimos App Store y en el buscador tecleamos Xcode, y se presenta la siguiente pantalla (Figura III - 8):

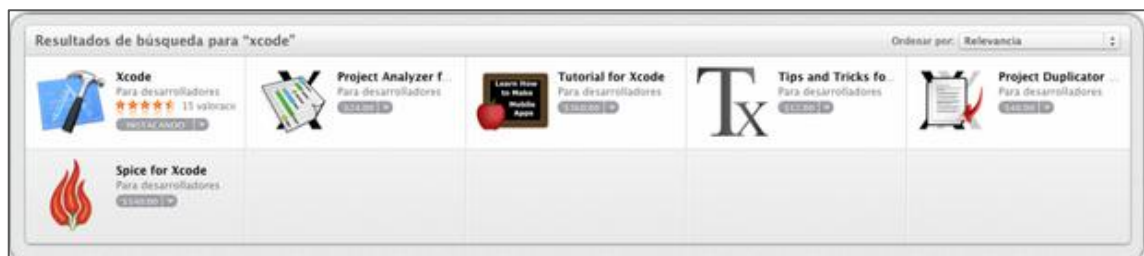


Figura III - 8 Descarga del Xcode

El IDE no tiene costo alguno, procedemos a instalarlo. El tamaño del IDE es superior a los 3GB.

Una vez finalizada la descarga, abrir el instalador de Xcode que descargamos, a continuación aceptamos el acuerdo de licencia y esperamos a que se instalen todos los componentes necesarios.



Figura III - 9 Acuerdo de Licencia de Xcode

Al finalizar la instalación se nos indicará que el proceso termino de manera exitosa.

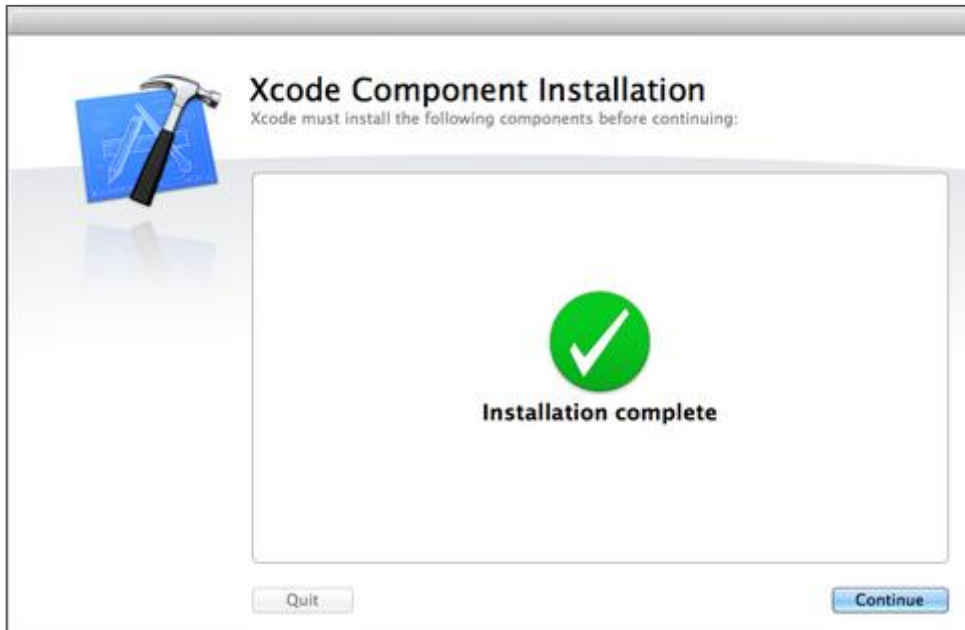


Figura III - 10 Instalación Completada

3.4 Determinación de los Parámetros de Comparación

Los parámetros que a continuación se define para el análisis entre las plataformas de desarrollo para dispositivos móviles inteligentes están basados en criterios generales de productividad en la implementación en el desarrollo de software⁵⁰ y criterios de los autores.

Los parámetros que se van a considerar para este análisis de las dos plataformas de desarrollo para dispositivos móviles inteligentes se encuentran en la TABLA III - II:

TABLA III - II Parámetros de Comparación

Parámetro	Descripción
1. Lenguajes de Programación	Este parámetro es muy importante porque nos permite determinar cuál es el lenguaje más adecuado para implementar aplicaciones de una manera ágil, de conocimiento de los desarrolladores y en que lenguajes se puede desarrollar en cada una de las plataformas.
2. Sistemas operativos	Se refiere a la disponibilidad de sistemas operativos en los que se puede montar una plataforma para el desarrollo de aplicaciones de dispositivos móviles inteligentes.
3. Entorno de Desarrollo Integrado	Este parámetro trata sobre las funcionalidades que tiene cada IDE para el desarrollo de aplicaciones nativas para dispositivos móviles

⁵⁰ Ing. Javier Salazar (Gerente de Proyectos de Software de la empresa Infoquality S.A.)

	inteligentes.
4. Interfaz Gráfica de Usuario	Se refiere a la representación de los distintos elementos de interfaz gráfica en las dos plataformas.
5. Persistencia de Datos	Se refiere a la gestión de la base de datos Sqlite que viene en el sistema operativo.
6. Acceso a servicios web	Se refiere a los estándares SOAP y REST utilizados en las dos plataformas para el consumo de servicios web.
7. Compatibilidad entre versiones	Se refiere a la compatibilidad de las aplicaciones que existe entre sus diferentes versiones de sistema operativo.
8. Facilidades de despliegue	Se refiere a la disponibilidad de emuladores y dispositivos físicos para desplegar una aplicación.

Los parámetros seleccionados anteriormente para el análisis comparativo se encuentran divididos en varios ítems que serán detallados a continuación:

TABLA III - III Indicadores del Parámetro de Comparación Lenguajes de Programación

Lenguajes de Programación	
Indicador	Descripción
Cantidad de lenguajes de programación	La variedad de lenguajes de programación para desarrollar aplicaciones con la misma plataforma.
Curva de aprendizaje de los lenguajes	Dificultad que el desarrollador tiene

de programación	en aprender el lenguaje de programación.
-----------------	--

TABLA III - IV Indicadores del Parámetro de Comparación Sistemas Operativos

Sistemas Operativos	
Indicador	Descripción
Cantidad de sistemas operativos soportados	Sistemas operativos soportados por la plataforma de desarrollo.
Cantidad de sistemas operativos gratuitos	Sistemas operativos gratuitos soportados por la plataforma de desarrollo.

TABLA III - V Indicadores del Parámetro de Comparación Interfaz Gráfica de Usuario

Interfaz Gráfica de Usuario	
Indicador	Descripción
Número de formas de definición de la estructura GUI	Las maneras en las que se definen los elementos en una interfaz de usuario.
Número de formas de fusión de código y diseño	Maneras para la vinculación de elementos GUI con el código.

TABLA III - VI Indicadores del Parámetro de Comparación Persistencia de datos

Persistencia de datos	
Indicador	Descripción
Facilidad de inserción de registros en Sqlite	Facilidad para implementar la inserción de registros a la base de datos.
Facilidad de actualización de registros en Sqlite	Facilidad para implementar la actualización de registros de una base de datos.

Facilidad de eliminación de registros en Sqlite	Facilidad para implementar la eliminación de registros de una base de datos.
Facilidad de consulta de registros en Sqlite	Facilidad para realizar consultas a la base de datos.

TABLA III - VII Indicadores del Parámetro de Comparación Acceso a Servicios Web

Acceso a Servicios Web	
Indicador	Descripción
Facilidad en el consumo de servicios web utilizando SOAP	Facilidad para acceder a métodos publicados en un servicio web mediante el estándar de comunicación SOAP.
Facilidad en el consumo de servicios web utilizando REST	Facilidad para acceder a métodos publicados en un servicio web mediante el estándar de comunicación REST.

TABLA III - VIII Indicadores del Parámetro de Comparación Entorno de Desarrollo Integrado

Entorno de Desarrollo Integrado	
Indicador	Descripción
Capacidad de ayuda inteligente	Ayuda inteligente con librerías propias del API.
Capacidad de Code Snippets	Capacidad para la escritura de piezas o partes de código.
Facilidad de depuración de aplicaciones	Depuración de aplicaciones dentro del IDE.
Capacidad del diseñador GUI visual	Ayuda en el diseño de interfaces de usuario

TABLA III - IX Indicadores del Parámetro de Comparación Compatibilidad entre Versiones

Compatibilidad entre Versiones	
Indicador	Descripción
Capacidad de manejo de versiones del SDK	Capacidad para manejar las versiones de APIS existentes para las plataformas de desarrollo.
Capacidad de compatibilidad de la aplicación para dispositivos de diferentes dimensiones de pantalla	Compatibilidad de las aplicaciones entre los dispositivos móviles inteligentes de diferentes dimensiones de pantalla.

TABLA III - X Indicadores del Parámetro de Comparación Facilidades de Despliegue

Facilidades de Despliegue	
Indicador	Descripción
Disponibilidad de emuladores	Emuladores disponibles en el SDK para el despliegue de una aplicación.
Facilidad para desplegar aplicaciones a un dispositivo físico	El procedimiento a realizar para desplegar una aplicación en un dispositivo físico.

3.5 AMBIENTE DE PRUEBAS

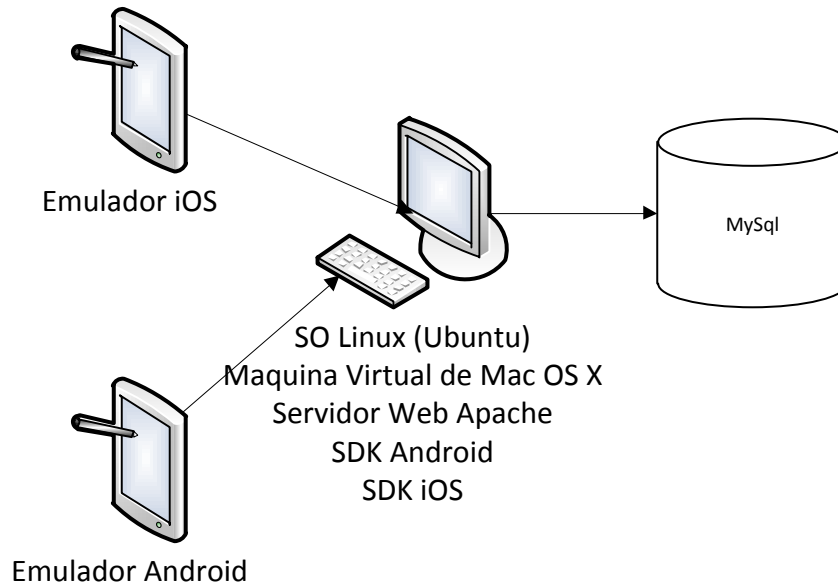


Figura III - 11 Instalación Completada

Los módulos de pruebas prácticas se desarrollaron en una computadora con sistema operativo Linux con la distribución Ubuntu, aquí se instaló la plataforma de desarrollo android, también se instaló VMWare Tools para instalar una máquina virtual de Mac OS X y la plataforma de desarrollo iOS. Cada módulo de prueba se testeó en emuladores de ambas plataformas.

3.6 DEMOSTRACIÓN DE PARÁMETROS DE COMPARACIÓN

3.6.1 Descripción de los Módulos de Prueba Prácticos

Módulo Prueba 1

Interfaz Gráfica de Usuario y Persistencia de Datos:

Este módulo consta de:

- a.) Manejo de un prototipo de agenda telefónica
- b.) Creación de una Base de Datos Sqlite que viene en el sistema operativo
- c.) Gestión de base de datos creada.

La gestión de contactos telefónicos será desarrollada tanto en la plataforma android como en la plataforma iOS para determinar los puntajes de cada plataforma en interfaz de usuario y persistencia de datos.

Módulo Prueba 2

Acceso a Servicios Web

Este módulo consta de:

- a.) Consumo de un servicio web que nos devuelve las categorías de platos existentes en el restaurante.
- b.) Consumo mediante soap.
- c.) Consumo mediante Rest con Json

El consumo del servicio web será desarrollado tanto en la plataforma android como en la plataforma iOS para determinar los puntajes de cada plataforma en el acceso a servicios web.

Módulo Prueba 3

Este módulo consta de:

- a.) Las capacidades que presta el Entorno de Desarrollo Integrado
- b.) La manera en que se maneja el control de versiones.
- c.) Las facilidades que brinda el SDK para desplegar una aplicación en un emulador o en un dispositivo físico.

Este módulo será desarrollado tanto en la plataforma android como en la plataforma iOS para determinar los puntajes de cada plataforma en el Entorno de Desarrollo Integrado, la compatibilidad entre versiones y las facilidades de despliegue.

3.6.2 Desarrollo de los Módulos de Prueba

3.6.2.1 Módulo Teórico de los Parámetros Lenguajes de Programación y Sistemas Operativos

Lenguajes de Programación

Android y iOS utilizan lenguajes en los que se pueden desarrollar aplicaciones para dispositivos móviles los cuales son:

Android

- Java (lenguaje por defecto)
- C/C++ (utilizando un NDK (Kit de Desarrollo Nativo))

iOS

- Objective-C (lenguaje por defecto)
- C

A continuación se presenta una tabla comparativa de los lenguajes de programación por defecto utilizados por estas dos plataformas de desarrollo:

TABLA III - XI Tabla Comparativa entre Java y Objective C

Característica	Java	Objective-C
Definición de Tipo de Datos	Tipado Estático: En este tipo de datos el tamaño que ocupa en memoria no puede variar durante la ejecución del programa. Es decir, una vez declarada una variable de un tipo determinado, a ésta se le asigna un espacio de memoria fijo, y este espacio no se podrá aumentar ni disminuir.	Tipado dinámico: Los tipos de datos son punteros. Este tipo te permite tener un mayor control sobre la gestión de memoria en los programas. Con ellos se puede manejar el tamaño de las variables en tiempo de ejecución, o sea, cuando el programa se está ejecutando.
Portabilidad	Puede ser ejecutado en diferentes entornos como (Windows, Linux y Mac OS X)	Puede ser ejecutado únicamente en MAC OS X
Curva de Aprendizaje	Posee una curva de aprendizaje muy corta.	La curva de aprendizaje para Objective-C es mayor en gran parte de los casos.
Sintaxis del Lenguaje	Sencilla. Por ejemplo para	Compleja. Por ejemplo para

	instancia un objeto en java se lo realiza asi: Object object = new Object();	instancia un objeto en Objective C se lo realiza asi: Object *object= [Object alloc]; [object init];
Gestión de Memoria	Utiliza un recolector de basura llamado Garbage collection que es un proceso el cual se encarga de limpiar la memoria.	Utiliza Automatic Reference Counting (ARC), este actua en tiempo de ejecución porque la memoria se libera en el momento que se deja de usar, no hay que esperar por un proceso que lo haga.
Comunidad de Desarrolladores	La comunidad de desarrolladores es muy grande, hay muchos libros, foros, blogs.	La comunidad de desarrolladores es mediana.

Sistemas Operativos

Un Sistema Operativo (SO) provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario. Entre los sistemas operativos compatibles para las plataformas Android y iOS son los siguientes:

ANDROID

- Windows (Propietario)
- Linux (i386) (Libre y Gratuito)
- Mac OS X (Intel) (Propietario)

iOS

- Mac OS X (Propietario)

3.6.2.2 Módulos de Prueba Prácticos del Parámetro Interfaz Gráfica de Usuario y Persistencia de Datos

3.6.2.2.1 Módulo 1 en Android

Para la construcción del módulo se va a utilizar los elementos de interfaz usuario del sdk de Android y el uso del IDE Eclipse, ya que estas nos permitirán construir interfaces graficas de usuario, para lo cual el sdk

proporciona componentes que permiten acceder y editar elementos desde el IDE.

Los pasos para el desarrollo del módulo se describe a continuación.

Creamos el proyecto para desarrollar el módulo el cual se llama ModuloPersistenciaDeDatos.

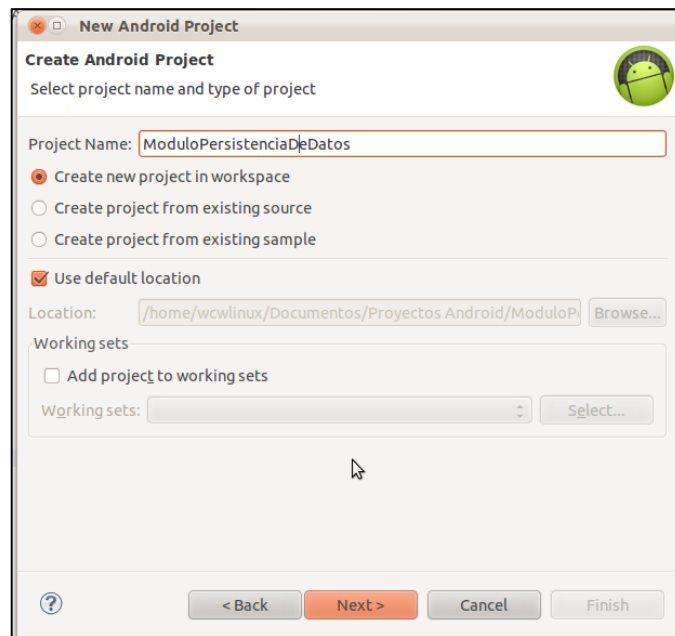


Figura III - 12 Creación de un Nuevo Proyecto

Escogemos la versión del sdk con el cual desarrollaremos el módulo.

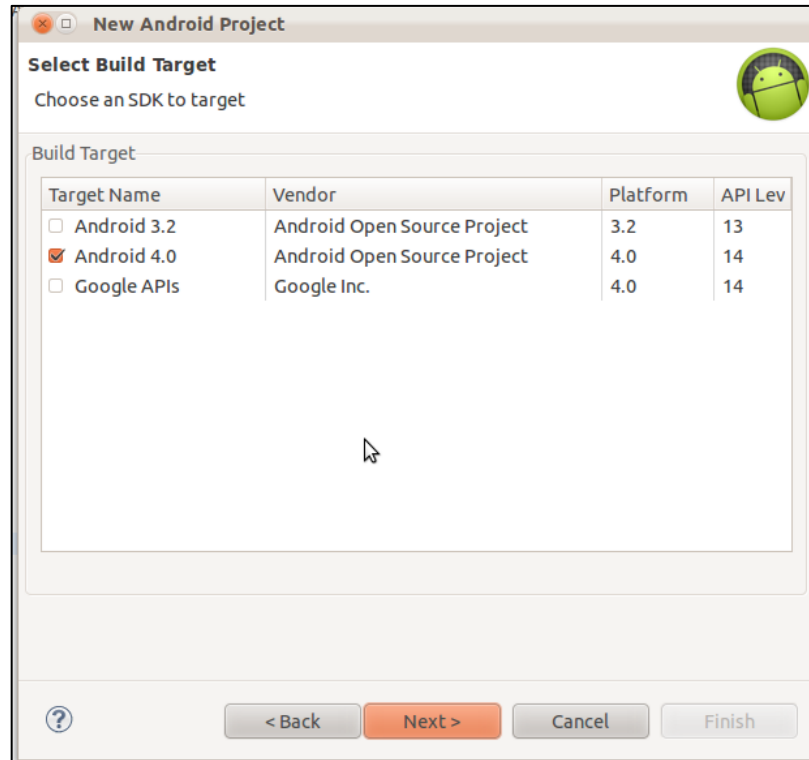


Figura III - 13 Selección de la versión de Android

Creamos la interfaz de usuario en un archivo XML el cual se llama main.xml que se encuentra en la carpeta de recursos res/layout, este archivo sirve para el manejo de la agenda telefónica con los elementos GUI del SDK de Android.

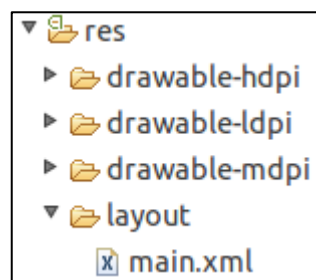


Figura III - 14 Directorios de carpetas para las interfaces en Android

Existen tres posibilidades para el diseño de las interfaces:

- Mediante interfaz gráfica del IDE.
- Mediante la definición de código XML.
- Mediante la definición código Java.

El IDE nos brinda la opción de arrastrar y soltar los elementos que formarán parte de la interfaz gráfica.

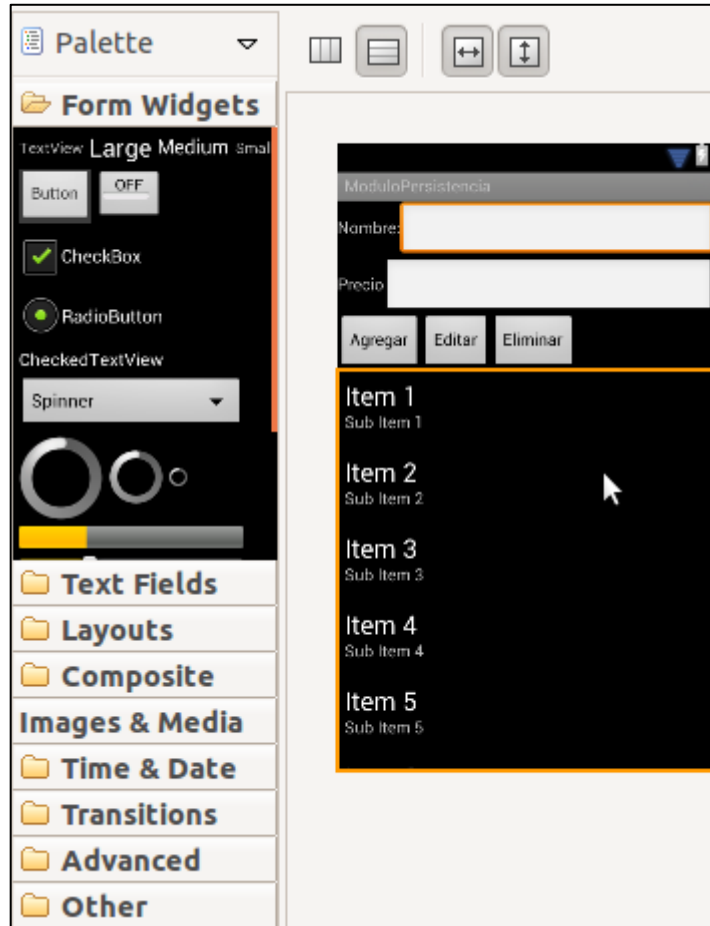


Figura III - 15 Paleta de elementos de la Interfaz Gráfica

También se puede definir la interfaz de usuario mediante código XML (Figura III - 16) que es muy similar a crear un documento HTML común.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nombre: " />

        <EditText
            android:id="@+id/name_editText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" >

            <requestFocus />
        </EditText>

    </LinearLayout>
```

Figura III - 16 Código XML de la Interfaz de Usuario

La última opción que brinda el SDK es definir la interfaz gráfica mediante código Java.

```
name_editText = new EditText(getBaseContext());
name_editText.setText("Hola Mundo");
```

Figura III - 17 Código Java de la Interfaz de Usuario

Para manipular los elementos de la interfaz gráfica creamos una actividad llamada MainActivity.java en la cual se carga el archivo main.xml.

```
setContentView(R.layout.main);
```

Figura III - 18 Llamada a la vista main

Para vincular y manejar los elementos de la interfaz gráfica se utiliza el método de la actividad llamado **findViewById()**.

```
name_editText = (EditText)findViewById(R.id.name_editText);
```

Figura III - 19 Vinculación del código Java con los elementos de la Vista

Existen dos maneras para manejar eventos de los elementos de la interfaz gráfica:

Definiendo la propiedad `android:onClick="<método>` con código XML del elemento e implementando el método en la actividad.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Agregar"
    android:onClick="insert"/>
```

Figura III - 20 Propiedad `onClick` en el código XML

Implementando los eventos de cada elemento mediante código Java.

```
new_button.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        name_editText.setText("");
        address_editText.setText("");
        phone_editText.setText("");
    }

});
```

Figura III - 21 Propiedad `onClick` en el código Java

Una vez definida la interfaz gráfica de usuario procedemos a implementar el código necesario para la gestión de la persistencia de datos de la agenda telefónica en la base de datos que viene incorporada en el sistema operativo llamada sqlite la cual nos permite la manipulación de datos mediante lenguaje sql.

Creación de la Base de Datos y tablas

Existen dos formas de crear una base de datos sqlite para android:

- La primera es generar previamente la base de datos mediante una herramienta para manipular bases de datos sqlite como por ejemplo el complemento para Firefox llamado sqlite manager.

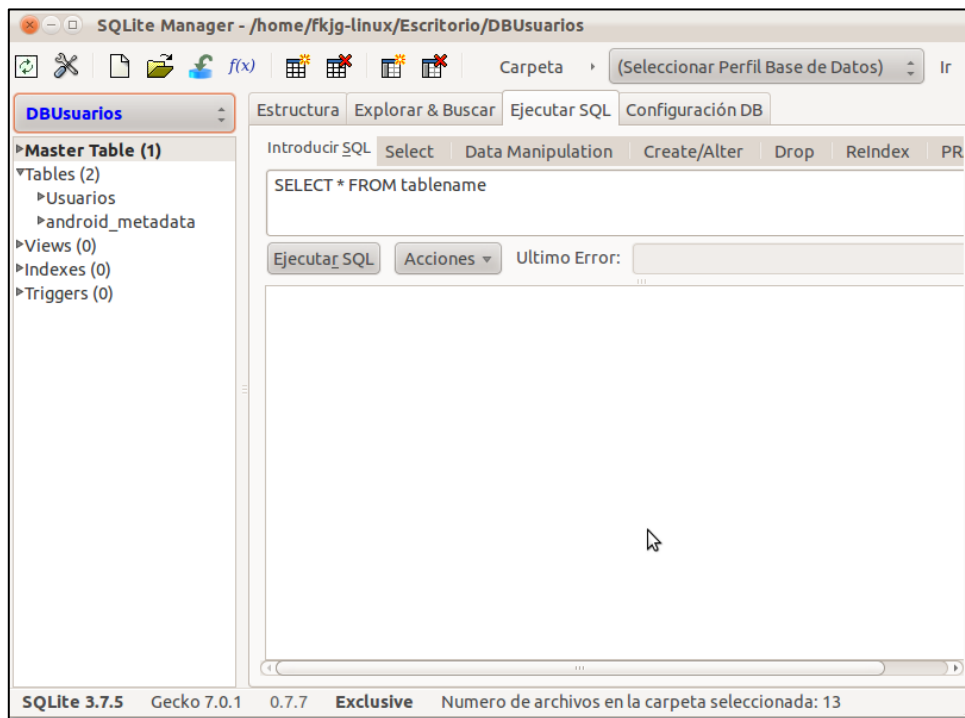


Figura III - 22 Creación de una Base de Datos con SQLite Manager

- La segunda es generar la base de datos con su tabla mediante código Java.

```
public class SQLiteHelperHandler extends SQLiteOpenHelper{

    String sqlCreate = "CREATE TABLE CONTACTS (ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, ADDRESS TEXT, PHONE TEXT)";

    public SQLiteHelperHandler(Context context, String name,
        CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(sqlCreate);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL("DROP TABLE IF EXISTS Contacts");
        db.execSQL(sqlCreate);
    }

}
```

Figura III - 23 Creación de una Base de Datos con Código Java

Para poder visualizar el archivo creado debemos ingresar por la vista file explorer y dirigirse a la siguiente ruta:

data/data/<paquete>/databases/<basededatos>

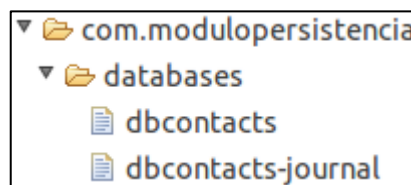


Figura III - 24 Directorio de la base de datos en el dispositivo

La base de datos que creamos se llama dbcontacts.sql, y la tabla se llama Contacts la cual contiene los campos (id, name, address, phone).

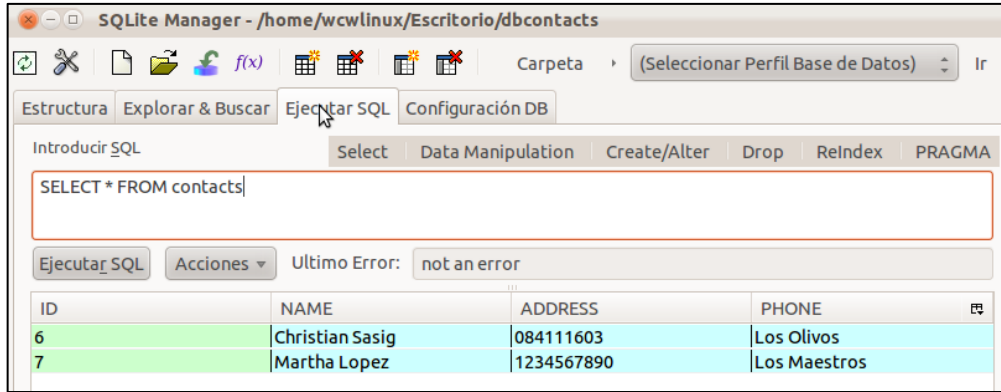


Figura III - 25 Registros de la tabla en el SQLite Manager

Inserción de Datos

Para insertar registros a la base de datos se utilizó el siguiente código.

```
public void insert(View v) {  
    String sql = "INSERT INTO CONTACTS (name, address, phone) "  
        + "VALUES ('" + name_editText.getText().toString() + " , '"  
        + phone_editText.getText().toString() + " , '"  
        + address_editText.getText().toString() + "')";  
    if (this.db != null) {  
        db.execSQL(sql);  
        name_editText.setText("");  
        address_editText.setText("");  
        phone_editText.setText("");  
    }  
    getContacts();  
}
```

Figura III - 26 Código Java para la inserción de registros

Para realizar una inserción de registros ingresar los datos en los cuadros de texto.

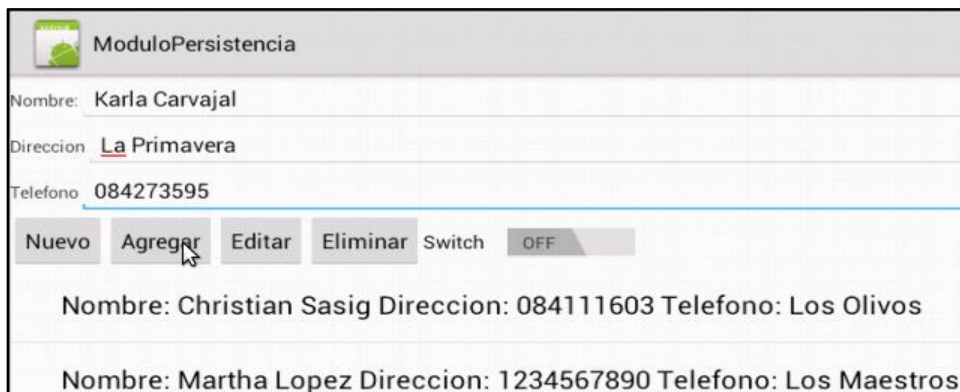


Figura III - 27 Pantalla de Ingreso de Datos

Después de dar click en el botón agregar se actualiza la lista de datos.

The screenshot shows a mobile application interface for managing contacts. At the top, there is a form with three input fields: 'Nombre:', 'Direccion', and 'Telefono'. Below the form is a horizontal bar with five buttons: 'Nuevo', 'Agregar', 'Editar', 'Eliminar', and 'Switch'. The 'Switch' button is currently in the 'OFF' position. Below the buttons, there is a list of three contacts, each displayed on a separate line with the following format: 'Nombre: [Name] Direccion: [Address] Telefono: [Phone]'. The contacts listed are: 1. Christian Sasig, Direccion: 084111603, Telefono: Los Olivos; 2. Martha Lopez, Direccion: 1234567890, Telefono: Los Maestros; 3. Karla Carvajal, Direccion: 084273595a, Telefono: La Primavera.

Figura III - 28 Listado de datos del contacto

Actualización de Datos

Para actualizar registros de la base de datos se utilizó el siguiente código.

```
public void update(View v) {  
    if (this.db != null) {  
  
        db.execSQL("UPDATE CONTACTS SET address='"  
            + address_editText.getText().toString() + "' , phone = '"  
            + phone_editText.getText().toString() + "' WHERE name='"  
            + name_editText.getText().toString() + "' ");  
    }  
    getContacts();  
}
```

Figura III - 29 Código Java para la actualización de registros

Para editar un registro seleccionar la fila y editar los campos.

This screenshot shows the same application interface as Figure III-28, but with the 'Editar' button highlighted by a mouse cursor. The form fields are filled with the data of the second contact: 'Nombre: Martha Lopez', 'Direccion: Los Maestros', and 'Telefono: 1234567890'. The list of contacts below remains the same as in the previous figure.

Figura III - 30 Pantalla de Actualización de Datos

Después de dar click en el botón editar se actualiza la lista de datos.

Nombre: Martha Lopez
Direccion: Los Maestros
Telefono: 1234567890

Nuevo Agregar Editar Eliminar Switch OFF

Nombre: Christian Sasig Direccion: 084111603 Telefono: Los Olivos
Nombre: Martha Lopez Direccion: Los Maestros Telefono: 1234567890
Nombre: Karla Carvajal Direccion: 084273595a Telefono: La Primavera

Figura III - 31 Listado de datos del contacto

Eliminación de Datos

Para eliminar los registros de la base de datos se utilizó el siguiente código.

```
public void delete(View v) {  
    if (this.db != null) {  
        db.execSQL("DELETE FROM CONTACTS WHERE name='"  
            + name_editText.getText().toString() + "'");  
        name_editText.setText("");  
        address_editText.setText("");  
        phone_editText.setText("");  
    }  
    getContacts();  
}
```

Figura III - 32 Código Java para la eliminación de registros

Para eliminar un registro seleccionar la fila que deseamos eliminar.

ModuloPersistencia

Nombre: Christian Sasig
Direccion: 084111603
Telefono: Los Olivos

Nuevo Agregar Editar Eliminar Switch ON

Nombre: Christian Sasig Direccion: 084111603 Telefono: Los Olivos
Nombre: Martha Lopez Direccion: Los Maestros Telefono: 1234567890
Nombre: Karla Carvajal Direccion: 084273595a Telefono: La Primavera

Figura III - 33 Búsqueda del Registro a eliminar

Después de dar click en el botón eliminar se actualiza la lista de datos.

Nombre:	<input type="text"/>
Direccion	<input type="text"/>
Telefono	<input type="text"/>
Nuevo Agregar Editar Eliminar Switch <input checked="" type="checkbox"/> ON	
Nombre: Martha Lopez Direccion: Los Maestros Telefono: 1234567890	
Nombre: Karla Carvajal Direccion: 084273595a Telefono: La Primavera	

Figura III - 34 Listado de datos del contacto

Consulta de Registros

Para presentar registros de la base de datos se utilizó el siguiente código.

```
public void getContacts() {
    Cursor c;

    ArrayList<Contact> contacts = new ArrayList<Contact>();
    try {
        c = db.rawQuery(" SELECT id, name, address, phone FROM Contacts",
            null);
        while (c.moveToNext()) {
            Contact contact = new Contact();
            contact.setId(c.getInt(0));
            contact.setName(c.getString(1));
            contact.setAddress(c.getString(2));
            contact.setPhone(c.getString(3));
            contacts.add(contact);
        }
    }
}
```

Figura III - 35 Código Java para recuperar datos de los registros

El resultado es el siguiente:

Nombre: Christian Sasig Direccion: 084111603 Telefono: Los Olivos
Nombre: Martha Lopez Direccion: 1234567890 Telefono: Los Maestros
Nombre: Karla Carvajal Direccion: 084273595a Telefono: La Primavera

Figura III - 36 Listado de datos del contacto

3.6.2.2.2 Módulo 1 en iOS

Para la construcción del módulo se va a utilizar los elementos de interfaz de usuario y la base de datos SQLite del sdk de iOS 4.3 y el uso del IDE Xcode, ya que estos nos permitirán construir interfaces graficas de usuario, para lo cual el sdk proporciona componentes que permiten acceder y editar elementos desde el IDE.

Los pasos para el desarrollo del módulo se describe a continuación.

Creamos un nuevo proyecto con la plantilla **View Based Application** y escogemos **iPad** en la parte producto.

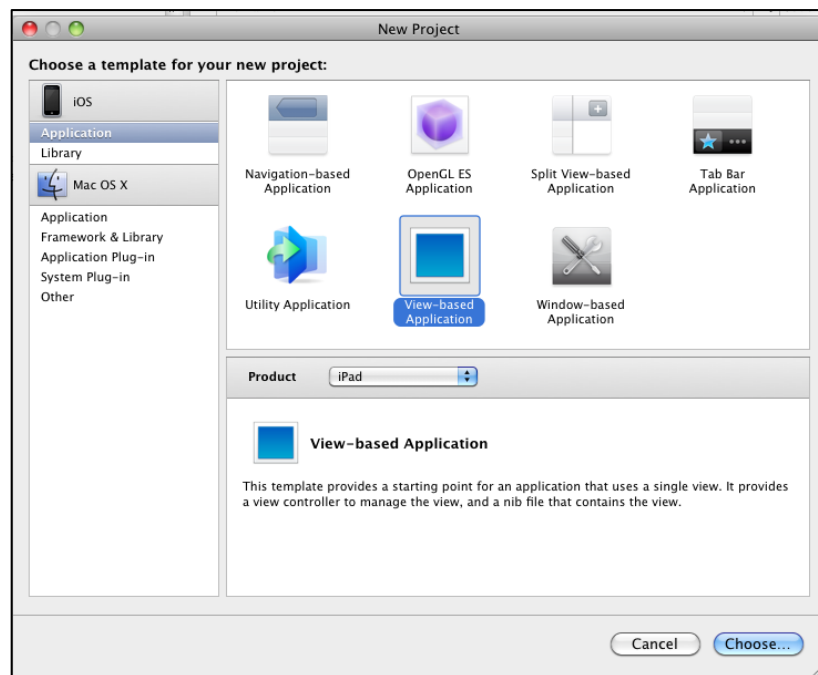


Figura III - 37 Pantalla de creación de un nuevo proyecto iOS con Xcode

Luego escribimos el nombre del proyecto el cual será ModuloPersistencia.

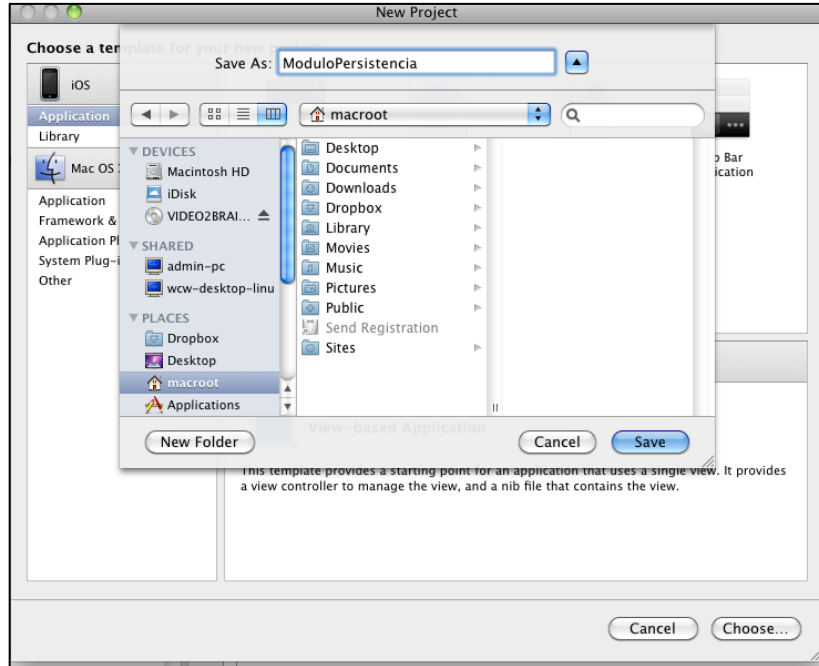


Figura III - 38 Establecer el nombre del proyecto

Creamos la interfaz de usuario en un archivo **.xib** el cual se llama **ModuloPersistenciaViewControler.xib** que se encuentra en la carpeta de recursos llamada **Resources**, este archivo sirve para el manejo de la agenda telefónica con los elementos GUI del SDK de iOS.

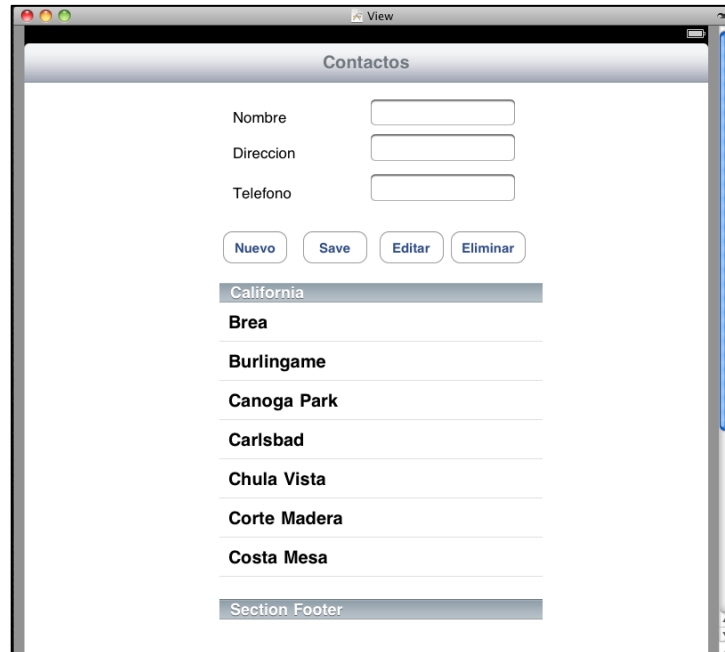


Figura III - 39 Interfaz Gráfica del módulo de persistencia

Existen dos posibilidades para el diseño de las interfaces:

- Mediante la herramienta de diseño de interfaces de usuario Interface Builder.
- Mediante la definición código Objective C.

El IDE nos brinda la opción de arrastrar y soltar los elementos que formarán parte de la interfaz gráfica.

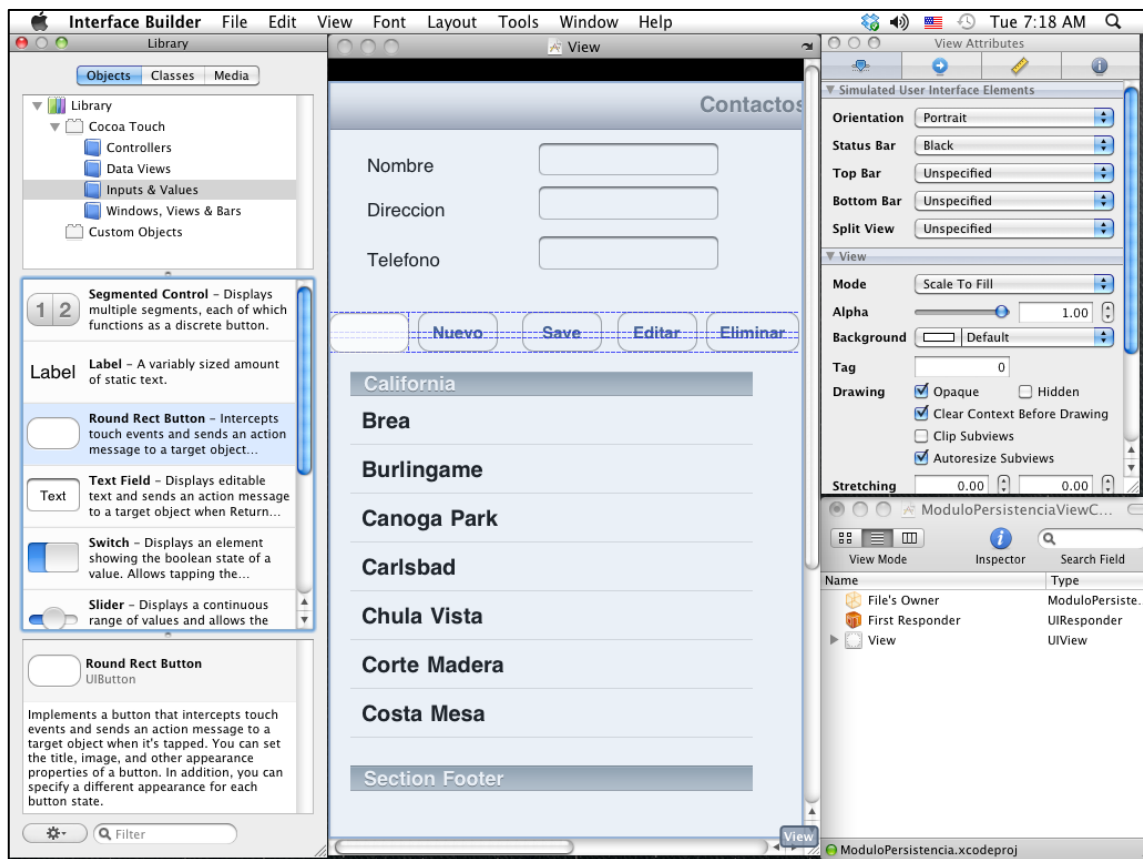


Figura III - 40 Herramientas para la Interfaz Gráfica

La última opción que brinda el SDK es definir la interfaz gráfica mediante código Objective C.

```
- (IBAction)buttonClicked:(id)sender
{
    NSLog(@"Contacto guardado");
}

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad {

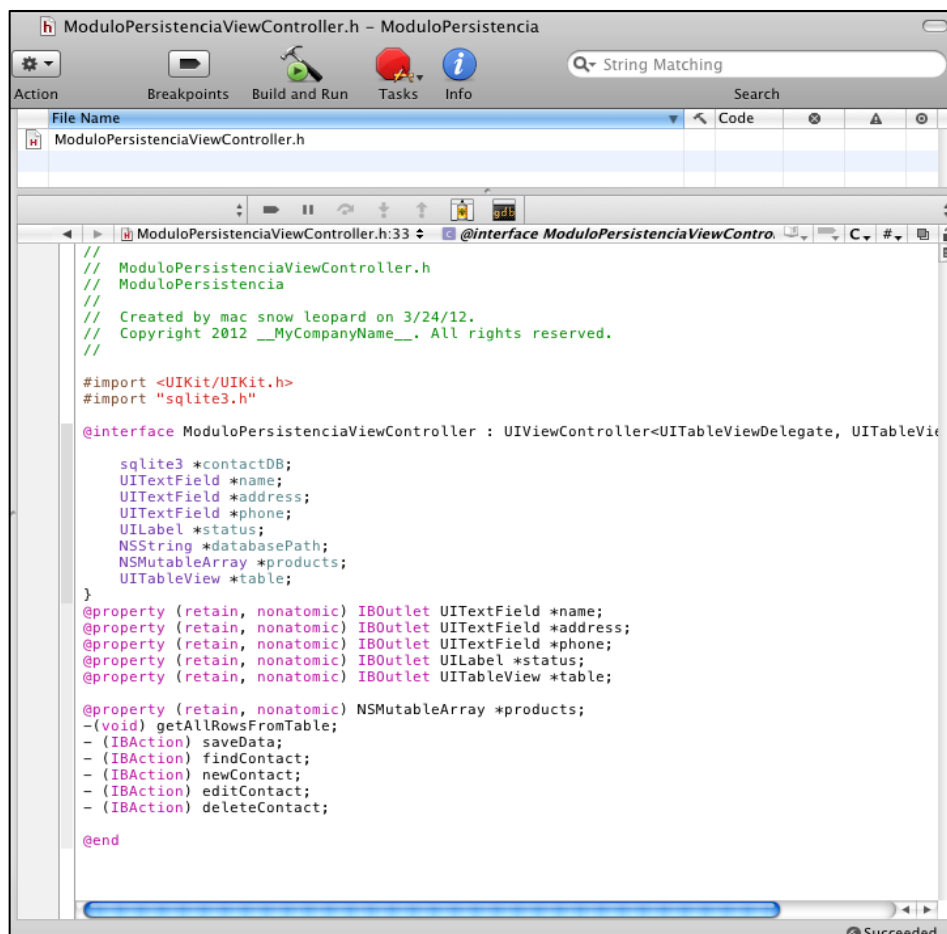
    UIButton *button = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    button.frame = CGRectMake(80, 50, 150, 40);

    [button setTitle:@"Guardar" forState:UIControlStateNormal];
    [button addTarget:self action:@selector(buttonClicked:) forControlEvents:UIControlEventTouchUpInside];

    [self.view addSubview:button];
}
```

Figura III - 41 Definición de la interfaz gráfica mediante código Objective-C

Para manipular los elementos de la interfaz gráfica creamos una clase controlador que herede de `UIViewController` **ModuloPersistenciaViewController.h** para la definición de propiedades y métodos y **ModuloPersistenciaViewController.m** para su implementación.



```
ModuloPersistenciaViewController.h - ModuloPersistencia

//
// ModuloPersistenciaViewController.h
// ModuloPersistencia
// Created by mac snow leopard on 3/24/12.
// Copyright 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "sqlite3.h"

@interface ModuloPersistenciaViewController : UIViewController<UITableViewDelegate, UITableViewDataSource>

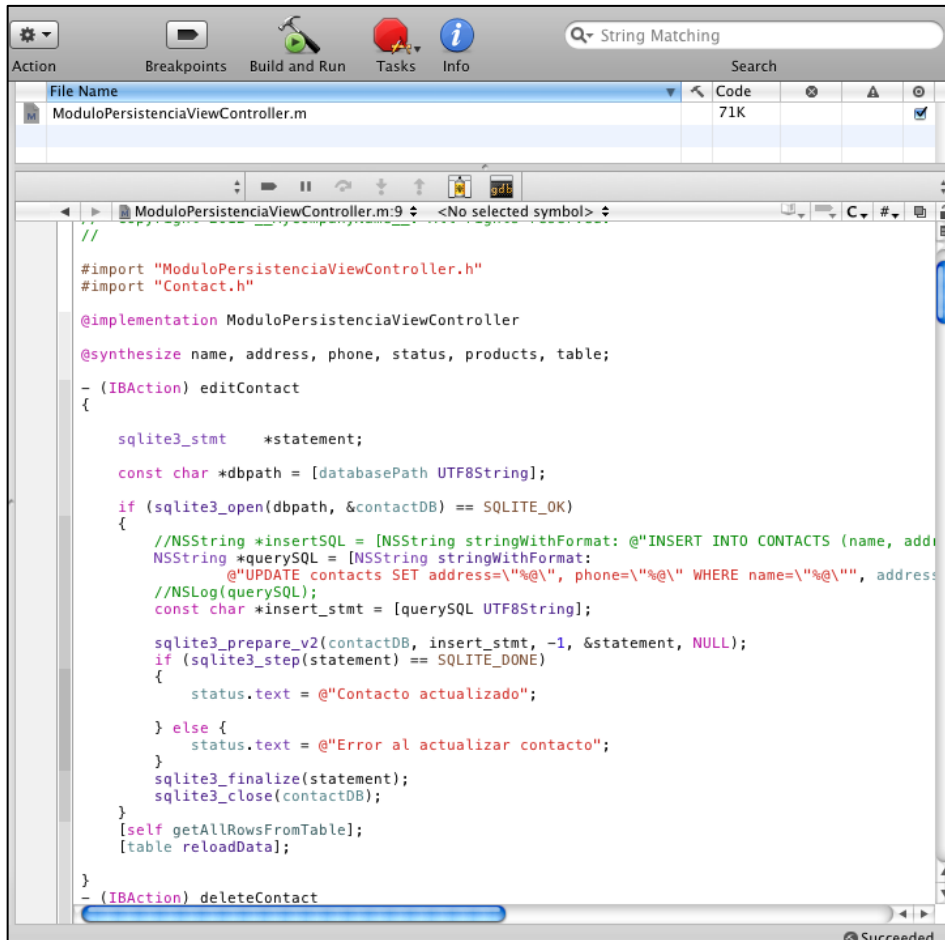
    sqlite3 *contactDB;
    UITextField *name;
    UITextField *address;
    UITextField *phone;
    UILabel *status;
    NSString *databasePath;
    NSMutableArray *products;
    UITableView *table;
}

@property (retain, nonatomic) IBOutlet UITextField *name;
@property (retain, nonatomic) IBOutlet UITextField *address;
@property (retain, nonatomic) IBOutlet UITextField *phone;
@property (retain, nonatomic) IBOutlet UILabel *status;
@property (retain, nonatomic) IBOutlet UITableView *table;

@property (retain, nonatomic) NSMutableArray *products;
-(void) getAllRowsFromTable;
- (IBAction) saveData;
- (IBAction) findContact;
- (IBAction) newContact;
- (IBAction) editContact;
- (IBAction) deleteContact;

@end
```

Figura III - 42 Creación de la clase ModuloPersistenciaViewController.h



```
//
#import "ModuloPersistenciaViewController.h"
#import "Contact.h"

@implementation ModuloPersistenciaViewController
@synthesize name, address, phone, status, products, table;

- (IBAction) editContact
{
    sqlite3_stmt *statement;
    const char *dbpath = [databasePath UTF8String];
    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        //NSString *insertSQL = [NSString stringWithFormat: @"INSERT INTO CONTACTS (name, address, phone) VALUES (%@, %@, %@)", name, address, phone];
        NSString *querySQL = [NSString stringWithFormat:
        @"UPDATE contacts SET address='%@\\', phone='%@\\' WHERE name='%@\\'", address, phone, name];
        //NSLog(querySQL);
        const char *insert_stmt = [querySQL UTF8String];
        sqlite3_prepare_v2(contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            status.text = @"Contacto actualizado";
        }
        else {
            status.text = @"Error al actualizar contacto";
        }
        sqlite3_finalize(statement);
        sqlite3_close(contactDB);
    }
    [self getAllRowsFromTable];
    [table reloadData];
}

- (IBAction) deleteContact
```

Figura III - 43 Creación de la clase ModuloPersistenciaViewController.m

Para la vinculación entre elementos de la interfaz y las propiedades definidas en la clase ModuloPersistencia.h (IBOutlet) se realiza visualmente mediante la herramienta Interface Builder.

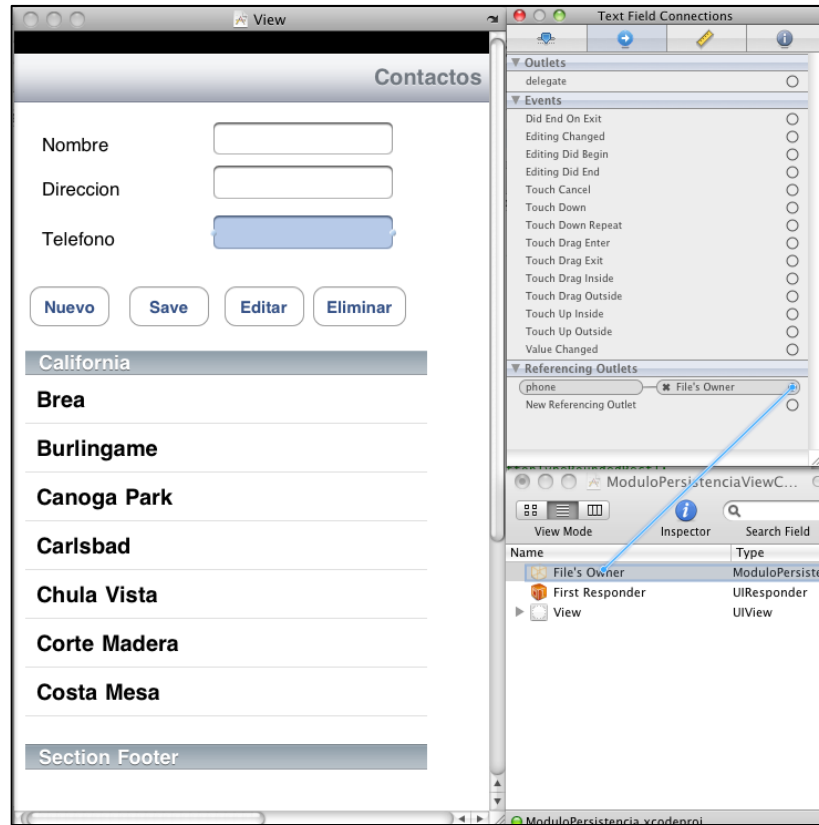


Figura III - 44 Vinculación de la interfaz de usuario con las propiedades de la clase

Existen dos maneras para manejar eventos de los elementos de la interfaz gráfica:

- Vinculando de manera visual entre los métodos definidos en la clase controlador y los eventos de cada elemento.

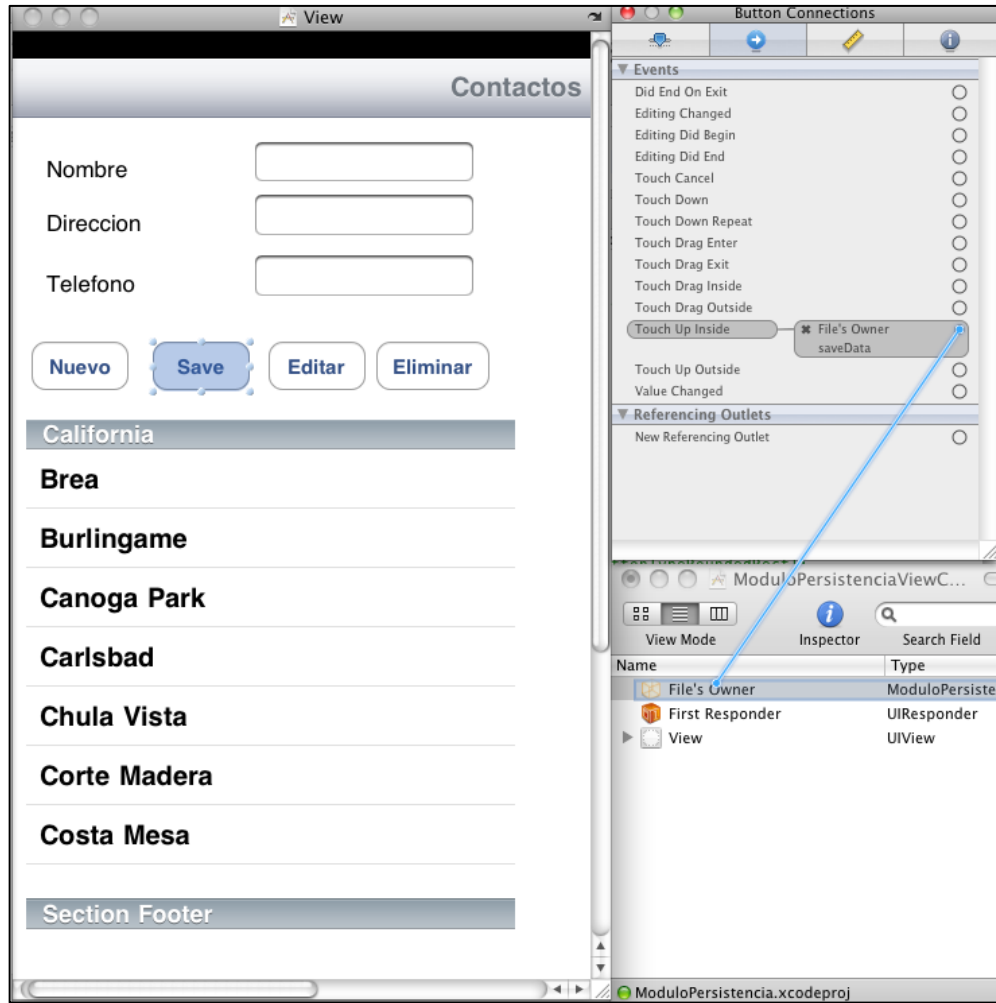


Figura III - 45 Vinculación de métodos de la clase con los eventos

- Implementando los eventos de cada elemento mediante código Objetivo C.

```
- (IBAction)buttonClicked:(id)sender
{
    NSLog(@"Contacto guardado");
}

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad {

    UIButton *button = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    button.frame = CGRectMake(80, 50, 150, 40);

    [button setTitle:@"Guardar" forState:UIControlStateNormal];
    [button addTarget:self action:@selector(buttonClicked:) forControlEvents:UIControlEventTouchUpInside];

    [self.view addSubview:button];
}
```

Figura III - 46 Implementación de eventos mediante código Objective-C

Una vez definida la interfaz gráfica de usuario procedemos a implementar el código necesario para la gestión de la persistencia de datos de la agenda telefónica en la base de datos que viene incorporada en el sistema operativo llamada sqlite la cual nos permite la manipulación de datos mediante lenguaje sql.

Creación de la Base de Datos y tablas

Existen dos formas de crear una base de datos sqlite para iOS:

- La primera es generar previamente la base de datos mediante una herramienta para manipular bases de datos sqlite como por ejemplo el complemento para Firefox llamado sqlite manager.

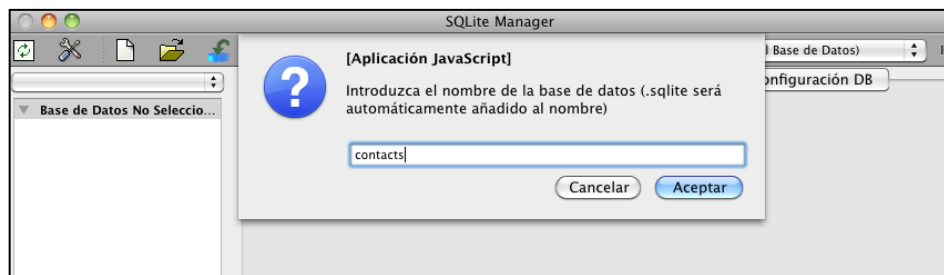


Figura III - 47 Creación de base de datos con SQLite Manager

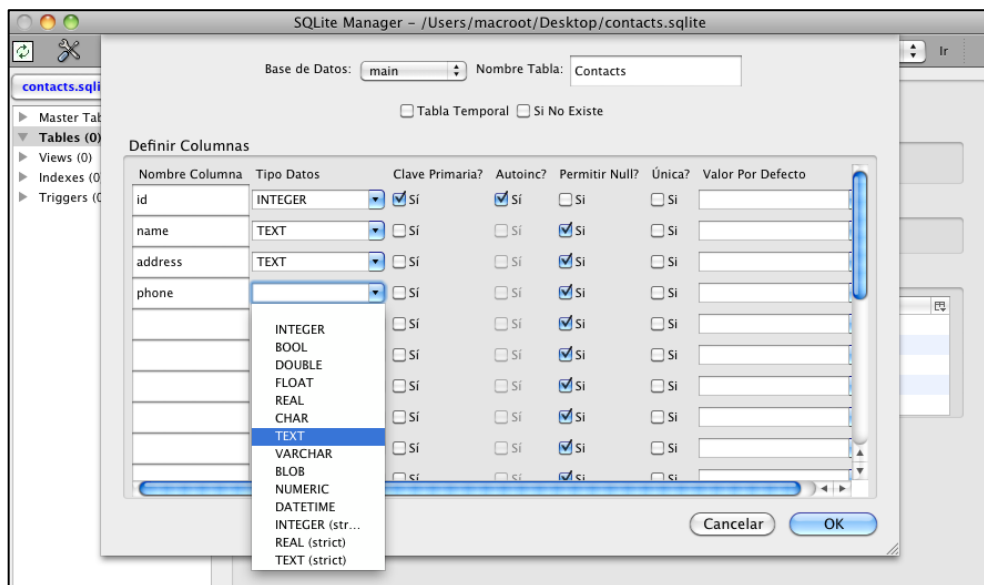


Figura III - 48 Creación de las tablas de la base de datos

- La segunda es generar la base de datos con su tabla mediante código Objective C.

```
/* Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
(void)viewDidLoad {
    NSString *docsDir;
    NSArray *dirPaths;

    dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
    docsDir = [dirPaths objectAtIndex:0];
    databasePath = [[NSString alloc] initWithString:[docsDir stringByAppendingPathComponent:@"contacts.db"]];

    NSFileManager *filemgr = [NSFileManager defaultManager];

    if ([filemgr fileExistsAtPath: databasePath] == NO)
    {
        const char *dbpath = [databasePath UTF8String];

        if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
        {
            char *errMsg;
            const char *sql_stmt = "CREATE TABLE IF NOT EXISTS CONTACTS (ID INTEGER PRIMARY KEY AUTOINCREMENT,
            NAME TEXT, ADDRESS TEXT, PHONE TEXT)";

            if (sqlite3_exec(contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK)
            {
                status.text = @"Error al crear la base de datos";
            }

            sqlite3_close(contactDB);
        } else {
            status.text = @"Error al crear o abrir la base de datos";
        }
    }
    [self getAllRowsFromTable];
    [filemgr release];
    [super viewDidLoad];
}
```

Figura III - 49 Creación de las tablas de la base de datos mediante código Objective-C

Para poder visualizar el archivo de base de datos creado debemos ingresar a la ruta:

~/Library/Application/Support/iPhone/Simulator/4.0/Applications/<App_ID>/Documents/<basededatos>

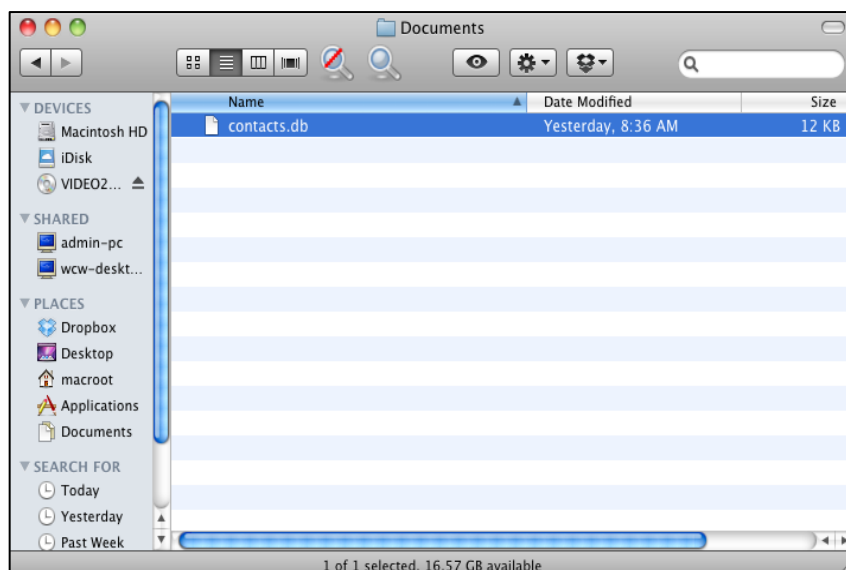


Figura III - 50 Archivo de la Base de Datos contacts.db

La base de datos que creamos se llama contacts.db, y la tabla se llama Contacts la cual contiene los campos(id, name, address, phone).

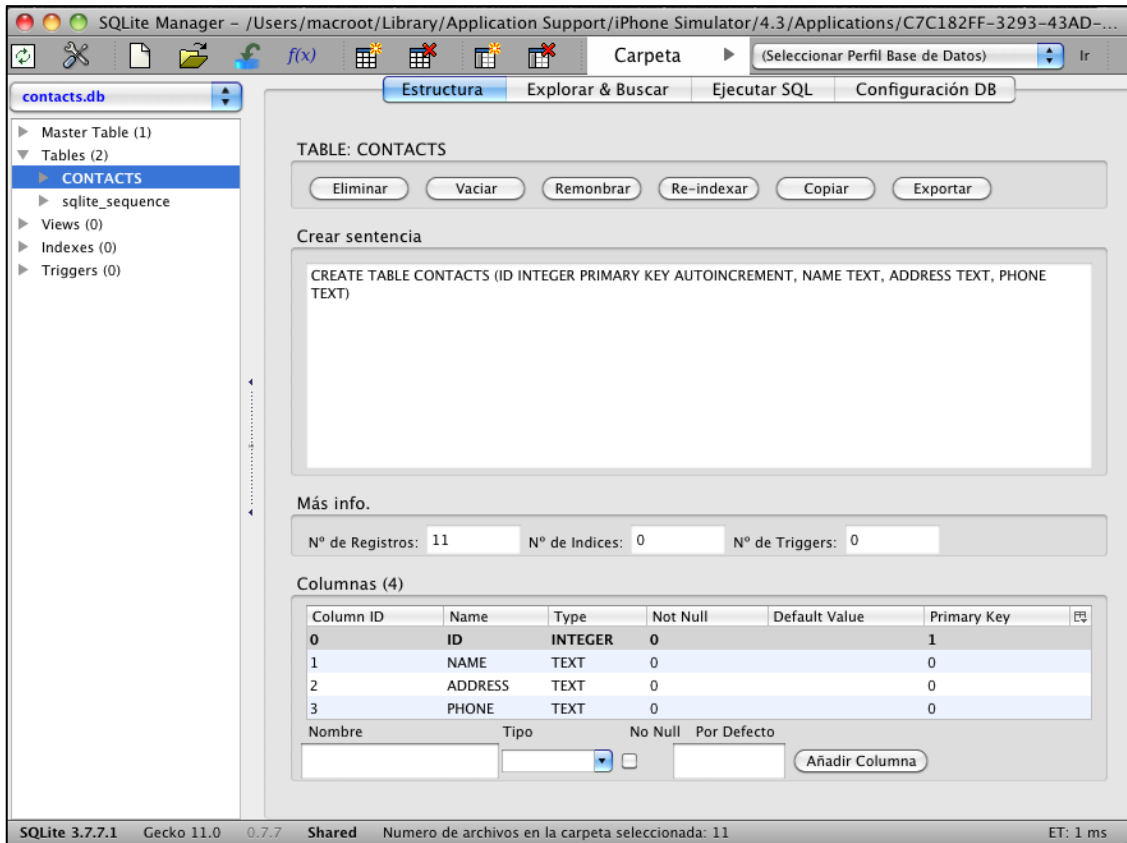


Figura III - 51 Tabla de la Base de Datos Contacts

Inserción de Datos

Para insertar registros a la base de datos se utilizó el siguiente código.

```
-(void) saveData
{
    sqlite3_stmt *statement;
    const char *dbpath = [databasePath UTF8String];
    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        NSString *insertSQL = [NSString stringWithFormat:@"INSERT INTO CONTACTS
        (name, address, phone) VALUES (\\%@\\", \\%@\\", \\%@\\")", name.text, address.text, phone.text];

        const char *insert_stmt = [insertSQL UTF8String];

        sqlite3_prepare_v2(contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            status.text = @"Contacto anadido";
            name.text = @"";
            address.text = @"";
            phone.text = @"";
        } else {
            status.text = @"Error al anadir contacto";
        }
        sqlite3_finalize(statement);
        sqlite3_close(contactDB);
    }
    [self getAllRowsFromTable];
    [table reloadData];
}
```

Figura III - 52 Código Objective-C para la inserción de registros

Para realizar una inserción de registros ingresar los datos en los cuadros de texto.

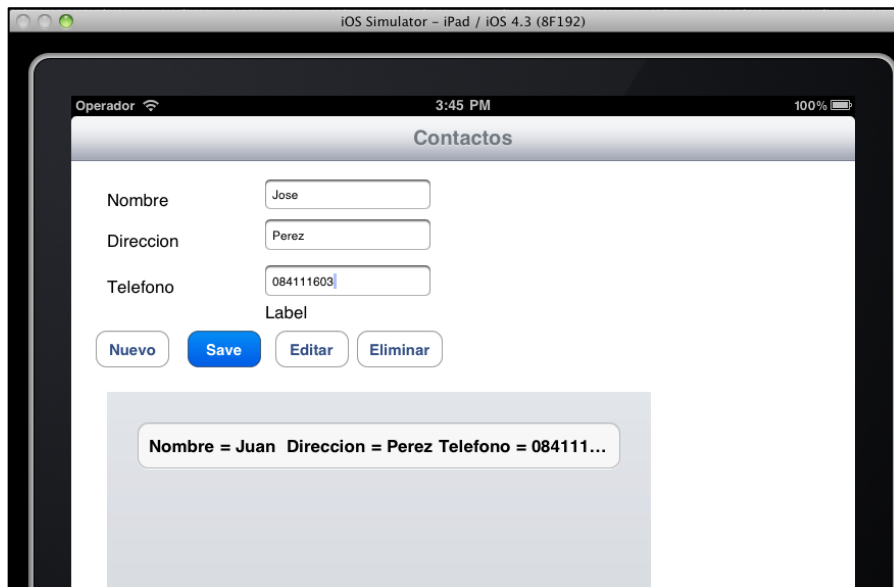


Figura III - 53 Pantalla para el ingreso de datos

Después de dar click en el botón agregar se actualiza la lista de datos.



Figura III - 54 Listado de Contactos

Actualización de Datos

Para actualizar registros de la base de datos se utilizó el siguiente código.

```
- (IBAction) editContact
{
    sqlite3_stmt *statement;

    const char *dbpath = [databasePath UTF8String];

    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        NSString *querySQL = [NSString stringWithFormat:@"UPDATE contacts
        SET address=\"%@\", phone=\"%@\" WHERE name=\"%@\"",
        address.text, phone.text, name.text];
        const char *insert_stmt = [querySQL UTF8String];

        sqlite3_prepare_v2(contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            status.text = @"Contacto actualizado";
        } else {
            status.text = @"Error al actualizar contacto";
        }
        sqlite3_finalize(statement);
        sqlite3_close(contactDB);
    }
    [self getAllRowsFromTable];
    [table reloadData];
}
```

Figura III - 55 Código Objective-C para la actualización de registros

Para editar un registro seleccionar la fila y editar los campos.



Figura III - 56 Pantalla de edición de Contactos

Después de dar click en el botón editar se actualiza la lista de datos.

Eliminación de Datos

Para eliminar los registros de la base de datos se utilizó el siguiente código.

```
- (IBAction) deleteContact
{
    sqlite3_stmt *statement;
    const char *dbpath = [databasePath UTF8String];

    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        //NSString *insertSQL = [NSString stringWithFormat:@"INSERT INTO CONTACTS (name, address, phone)
        VALUES ("%@%", "%%", "%%)", name.text, address.text, phone.text];
        NSString *querySQL = [NSString stringWithFormat:@"DELETE FROM contacts WHERE name=\"%@\\"", name.text];
        //NSLog(querySQL);
        const char *insert_stmt = [querySQL UTF8String];

        sqlite3_prepare_v2(contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            status.text = @"Contacto eliminado";
        } else {
            status.text = @"Error al eliminar contacto";
        }
        sqlite3_finalize(statement);
        sqlite3_close(contactDB);
    }
    [self getAllRowsFromTable];
    [table reloadData];
}
```

Figura III - 57 Código Objective-C para la eliminación de registros

Para eliminar un registro seleccionar la fila que deseamos eliminar.

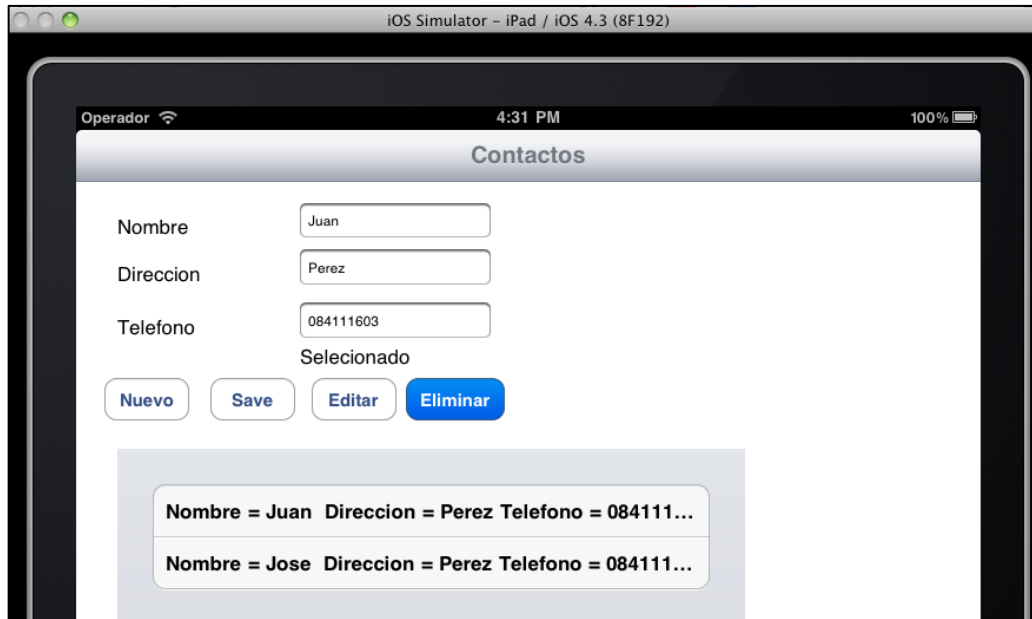


Figura III - 58 Búsqueda del registro a eliminar

Después de dar clic en el botón eliminar se actualiza la lista de datos.

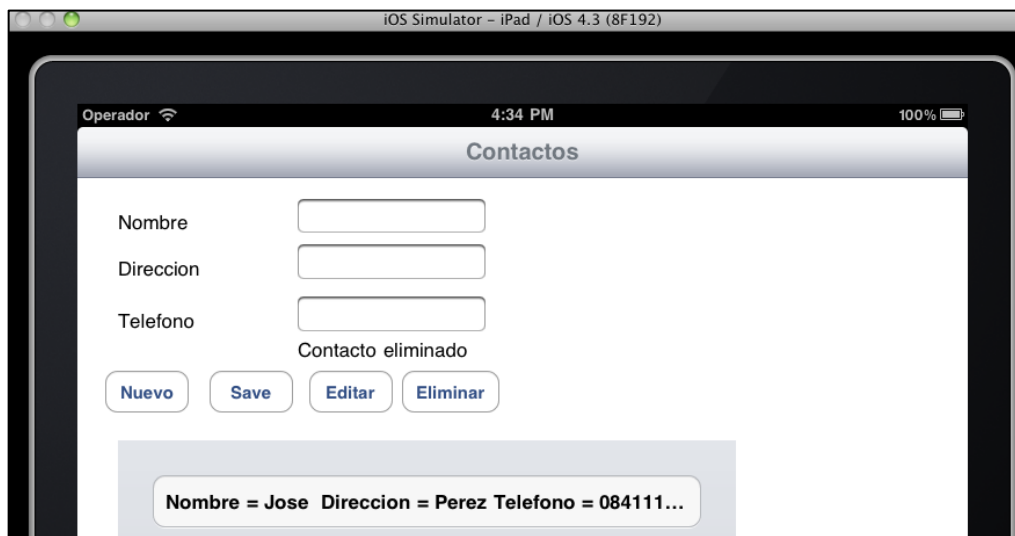


Figura III - 59 Resultado de la eliminación de registros

Consulta de Registros

Para presentar registros de la base de datos se utilizó el siguiente código.

```
- (void) FindContact
{
    const char *dbpath = [databasePath UTF8String];
    sqlite3_stmt *statement;

    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        NSString *querySQL = [NSString stringWithFormat:@"SELECT address, phone FROM contacts WHERE name=\"%@\"", name.text];
        const char *query_stmt = [querySQL UTF8String];

        if (sqlite3_prepare_v2(contactDB, query_stmt, -1, &statement, NULL) == SQLITE_OK)
        {
            if (sqlite3_step(statement) == SQLITE_ROW)
            {
                NSString *addressField = [[NSString alloc] initWithUTF8String:(const char *) sqlite3_column_text(statement, 0)];
                address.text = addressField;
                NSString *phoneField = [[NSString alloc] initWithUTF8String:(const char *) sqlite3_column_text(statement, 1)];
                phone.text = phoneField;
                status.text = @"Encontrado";
                [addressField release];
                [phoneField release];
            } else {
                status.text = @"No Encontrado";
                address.text = @"";
                phone.text = @"";
            }
            sqlite3_finalize(statement);
        }
        sqlite3_close(contactDB);
    }
}
```

Figura III - 60 Código Objective-C para la recuperación de registros

El resultado es el siguiente:

Nombre = Juan Direccion = Perez Telefono = 084111...
Nombre = Jose Direccion = Perez Telefono = 084111...

Figura III - 61 Listado de Contactos

3.6.2.3 Módulos de Prueba del Parámetro Acceso a Servicios Web

3.6.2.3.1 Módulo 2 en Android

Para la construcción del módulo vamos a utilizar dos mecanismos de comunicación que son SOAP y REST que permiten el consumo de servicios web, mediante la utilización de librerías y métodos que proporciona el propio SDK.

Los pasos para el desarrollo del módulo se describe a continuación.

Creamos el proyecto para desarrollar el módulo el cual se llama ModuloServicioWeb y seleccionamos la versión del SDK.

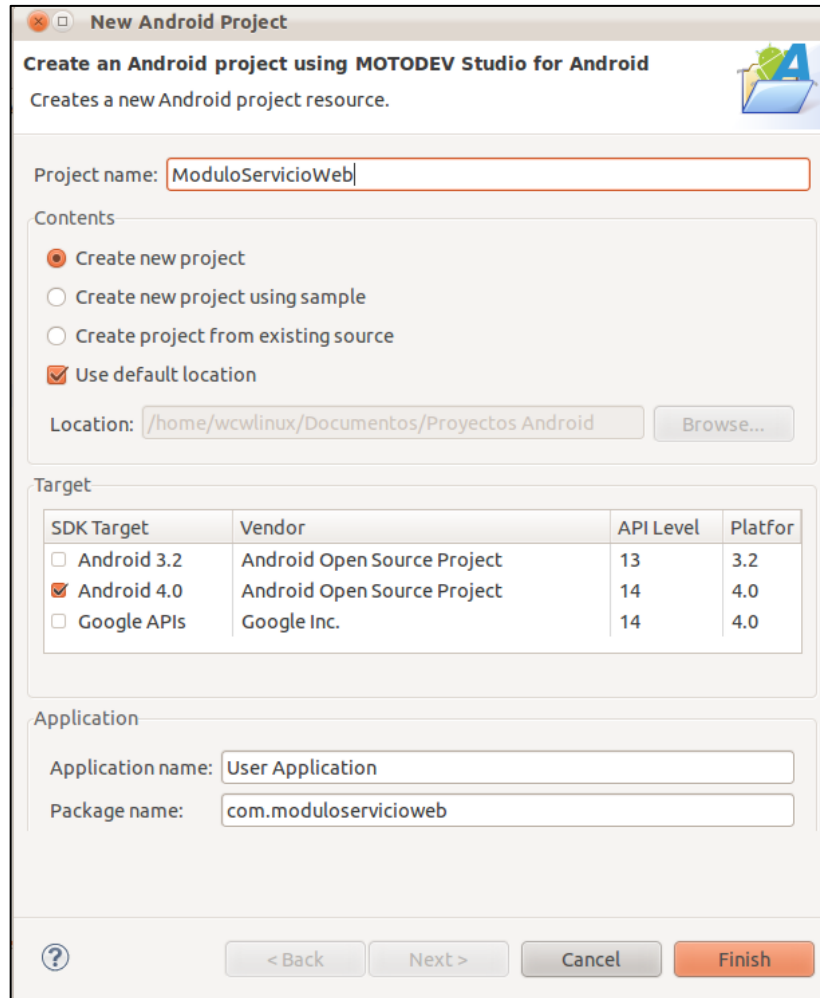


Figura III - 62 Creación del Módulo Servicio Web

Acceso a Servicios Web mediante REST

Para realizar el intercambio de datos mediante REST se utiliza el formato XML o JSON. Utilizaremos JSON para demostrar la práctica.

Utilizaremos el servicio web publicado en el sitio web para la administración del restaurante que nos devuelve las categorías de platos en formato JSON; el servicio web se llama get-menu-groups.

```
public function getMenuGroupsAction() {
    $request = $this->getRequest();
    $em = $this->getDoctrine()->getEntityManager();
    $menugroups = $em->getRepository('AdministrationBundle:MenuGroup')->findBy(array(),
        array('name' => 'ASC'));

    $menugroups_array = array();
    foreach ($menugroups as $menugroup) {
        $menugroup_array['id'] = $menugroup->getId();
        $menugroup_array['name'] = $menugroup->getName();
        $menugroup_array['description'] = $menugroup->getDescription();
        $menugroup_array['status'] = $menugroup->getStatus();
        $menugroups_array[] = array('menugroup' => $menugroup_array);
    }

    echo json_encode(array('menugroups' => $menugroups_array));
    exit;
}
```

Figura III - 63 Creación del Servicio Web

Generamos un AsyncTask que permite realizar procesos en background para que otras tareas puedan ejecutarse y no se cuelgue la aplicación en este caso el consumo del servicio web es un proceso en background.

```
private class GetMenuGroupsJSON extends AsyncTask<String, Void, ArrayList<MenuGroup> > {
    @Override
    protected ArrayList<MenuGroup> doInBackground(String... url) {
        ArrayList<MenuGroup> menugroups = new ArrayList<MenuGroup>();
        HttpClient httpClient = new DefaultHttpClient();
        HttpGet get = new HttpGet(url[0]);

        get.setHeader("content-type", "application/json");
        HttpResponse resp;

        try {
            resp = httpClient.execute(get);
            String respStr = EntityUtils.toString(resp.getEntity());

            JSONObject respJSON = new JSONObject(respStr);
            String data_menugroups = respJSON.getString("menugroups");
            JSONArray json_array = new JSONArray(data_menugroups);

            for (int i = 0; i < json_array.length(); i++) {
                JSONObject userObj = json_array.getJSONObject(i);
                String userStr = userObj.getString("menugroup");
                JSONObject item = new JSONObject(userStr);

                MenuGroup menugroup = new MenuGroup();
                menugroup.setId(item.getInt("id"));
                menugroup.setName(item.getString("name"));
                menugroup.setDescription(item.getString("description"));
                menugroup.setStatus(item.getString("status"));
                menugroups.add(menugroup);
            }
        }
    }
}
```



```
    } catch (ClientProtocolException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (JSONException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
return menugroups;  
}
```

Figura III - 64 Código Java del Consumo de un servicio web con JSON

Ejecutamos la función.

```
ListView menugroups_listView;  
Button view_data_button;  
  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    menugroups_listView = (ListView)findViewById(R.id.menugroups_listView);  
    view_data_button = (Button)findViewById(R.id.view_data_button);  
  
    view_data_button.setOnClickListener(new OnClickListener() {  
  
        public void onClick(View v) {  
  
            new GetMenuGroupsJSON().execute("http://192.168.137.101/sigres/web/app_dev.php/menugroup/json/get-menu-groups");  
  
        }  
    });  
}
```

Figura III - 65 Código Java de la ejecución del método

A continuación asignamos permisos de acceso a internet en el archivo AndroidManifest.xml.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.moduloservicioweb"  
    android:versionCode="1"  
    android:versionName="1.0">  
    <uses-sdk android:minSdkVersion="14" />  
    <uses-permission android:name="android.permission.INTERNET"/>  
    <application android:icon="@drawable/ic_launcher" android:label="@string/app_name"  
        <activity android:name="MainActivity"  
            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>
```

Figura III - 66 Permiso de Internet en el archivo AndroidManifest.xml

El resultado es el siguiente.

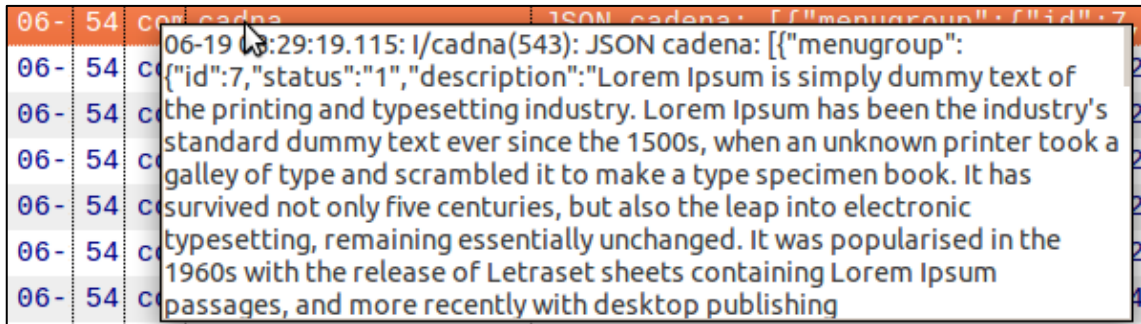


Figura III - 67 Resultado del consumo de datos mediante JSON

Acceso a Servicios Web mediante SOAP

Para realizar el intercambio de datos mediante SOAP se utiliza el formato XML. Utilizaremos el servicio web publicado en el sitio web para la administración del restaurante que nos devuelve las categorías de platos en formato XML; el método publicado en el servicio web se llama get-categories.

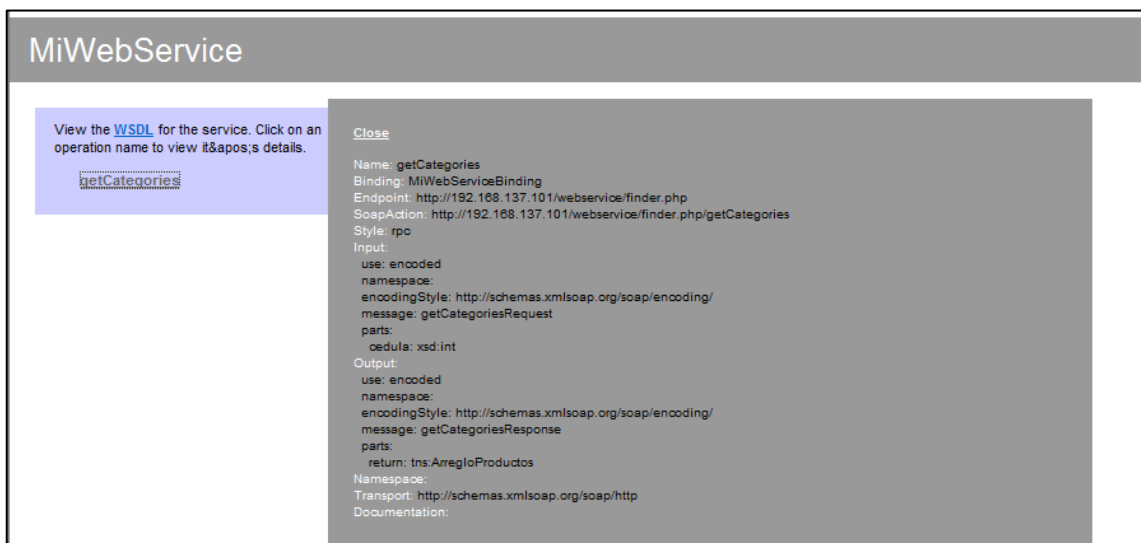


Figura III - 68 Servicio Web GetCategories

Llamamos al servicio web y lo consumimos de la siguiente manera

```
String NAMESPACE = "http://192.168.137.101/";
String URL="http://192.168.137.101/webservice/finder.php";
String METHOD_NAME = "getCategories";
String SOAP_ACTION = "http://192.168.137.101/webservice/finder.php/getCategories";

SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.setOutputSoapObject(request);
HttpTransportSE transporte = new HttpTransportSE(URL);

ArrayList<MenuGroup> menugroups = new ArrayList<MenuGroup>();

try
{
    transporte.call(SOAP_ACTION, envelope);
    Vector resSoap =(Vector)envelope.getResponse();
    Log.e("xml", envelope.getResponse().toString());
}
```

Figura III - 69 Código Java del consumo de datos mediante SOAP

Los datos obtenidos del consumo del servicio web son mostrados mediante una ventana de logs.

```
loservicio: xml org.ksoap2.serialization.SoapSerializationEnvelope@41071648
loservicio: gralloc_goldfish Emulator w 04-13 16:58:05.039: E/xml(731): [Productos{id=1; name=Mariscos y
loservicio: xml org.ksoap2 Emulator w Pescado; description=Lorem Ipsum is simply dummy text of the printing
loservicio: gralloc_goldfish Emulator w and typesetting industry. Lorem Ipsum has been the industry's standard
loservicio: xml [Productos dummy text ever since the 1500s, when an unknown printer took a galley of
type and scrambled it to make a type specimen book. It has survived not
only five centuries, but also the leap into electronic typesetting, remaining
essentially unchanged. It was popularised in the 1960s with the release of
Letraset sheets containing Lorem Ipsum passages, and more recently with
desktop publishing so; status=1; ], Productos{id=2; name=Carnes;
```

Figura III - 70 Resultado del consumo de datos mediante SOAP

3.6.2.3.2 Módulo 2 en iOS

Para la construcción del módulo vamos a utilizar dos mecanismos de comunicación que son SOAP y REST que permiten el consumo de servicios web, mediante la utilización de librerías y métodos que proporciona el propio SDK.

Los pasos para el desarrollo del módulo se describe a continuación.

Creamos un nuevo proyecto con la plantilla **View Based Application** y escogemos **iPad** en la parte producto.

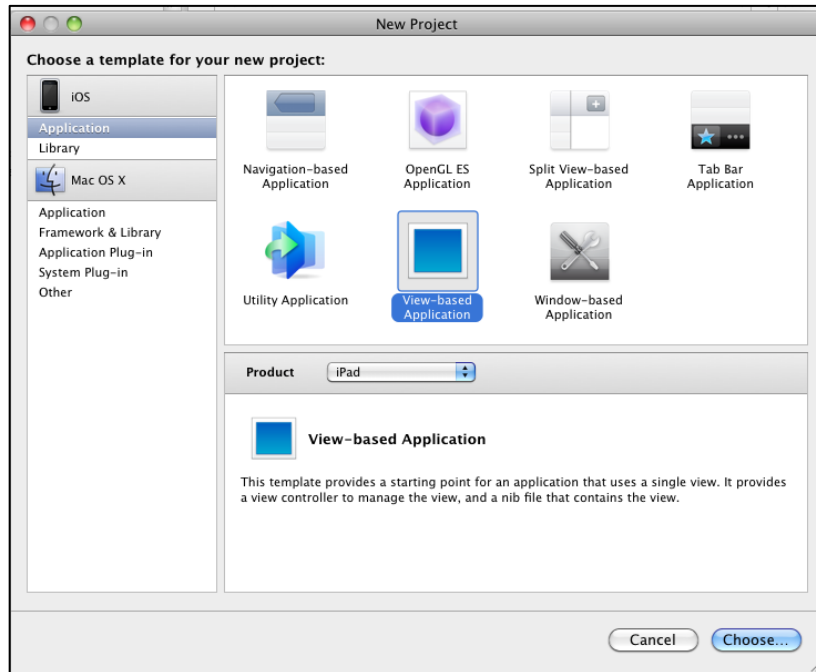


Figura III - 71 Creación de un Proyecto en Xcode

Luego escribimos el nombre del proyecto el cual será **ModuloServiciosWeb**.

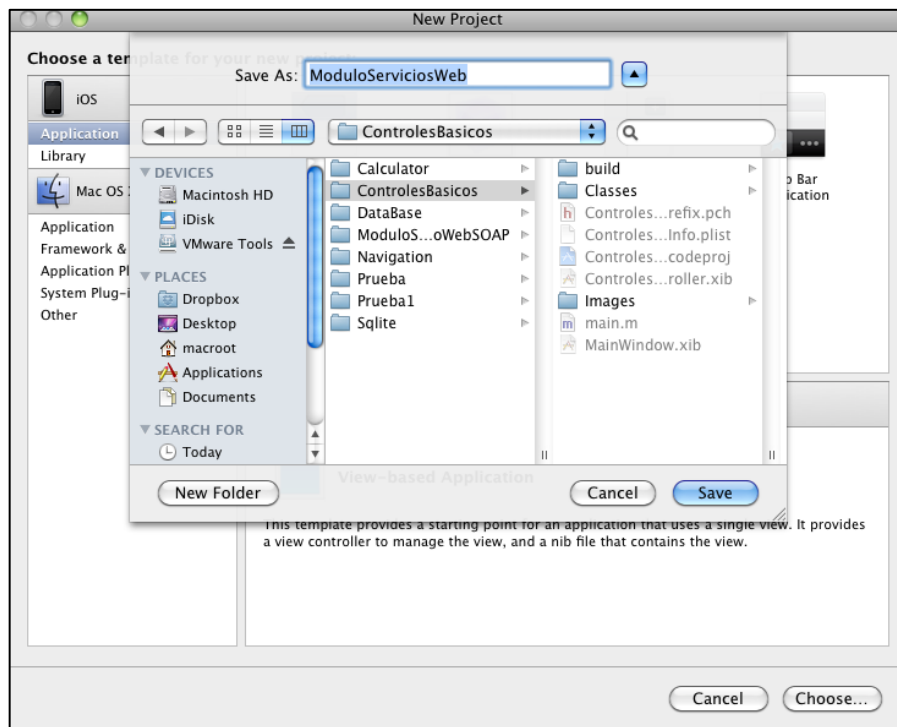


Figura III - 72 Ventana de Creación del Proyecto en XCode

Acceso a Servicios Web mediante REST

Para realizar el intercambio de datos mediante REST se utiliza el formato XML o JSON. Utilizaremos JSON para demostrar la práctica.

Utilizaremos el servicio web publicado en el sitio web para la administración del restaurante que nos devuelve las categorías de platos en formato JSON; el servicio web se llama get-menu-groups.

```
public function getMenuGroupsAction() {
    $request = $this->getRequest();
    $em = $this->getDoctrine()->getEntityManager();
    $menugroups = $em->getRepository('AdministrationBundle:MenuGroup')->findBy(array(),
        array('name' => 'ASC'));

    $menugroups_array = array();
    foreach ($menugroups as $menugroup) {
        $menugroup_array['id'] = $menugroup->getId();
        $menugroup_array['name'] = $menugroup->getName();
        $menugroup_array['description'] = $menugroup->getDescription();
        $menugroup_array['status'] = $menugroup->getStatus();
        $menugroups_array[] = array('menugroup' => $menugroup_array);
    }

    echo json_encode(array('menugroups' => $menugroups_array));
    exit;
}
```

Figura III - 73 Código del Servicio web con PHP

Para el uso de JSON, usamos las librerías de JSONKit que incluyen dos archivos JSONKit.h y JSONKit.m que se encuentran en la dirección <https://github.com/johnejang/JSONKit>.

Añadimos los archivos al proyecto (Figura III - 74).

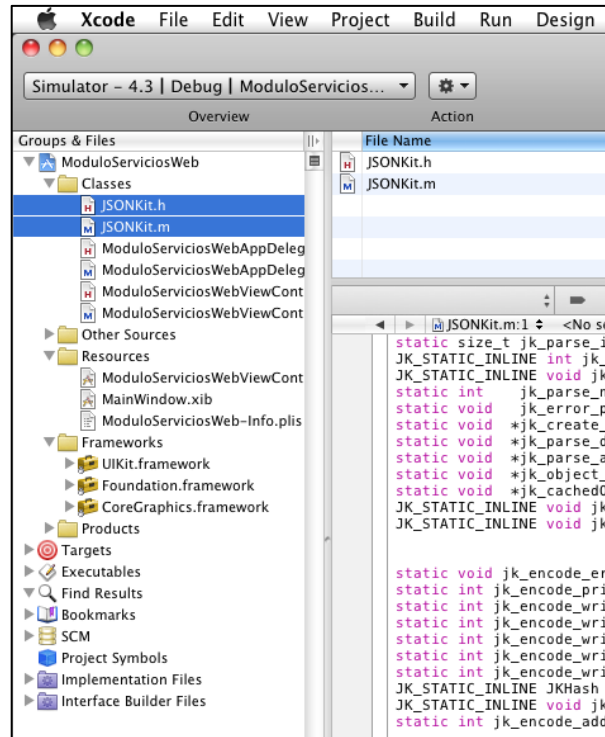


Figura III - 74 Librerías de JSON JSONKit.h y JSONKit.m

Para consumir el servicio web usamos la clase `NSURLRequest` y `NSURLConnection` como podemos observar en el siguiente gráfico.

```
- (void)viewDidLoad {
    [super viewDidLoad];

    datos = [[NSMutableData data] retain];
    NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL URLWithString:
@"http://192.168.137.101/sigres/web/app_dev.php/menugroup/json/get-menu-groups"]];

    [[NSURLConnection alloc] initWithRequest:request delegate:self];
}

- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response {
    [datos setLength:0];
}

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data {
    [datos appendData:data];
}

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error {
    NSLog(@"Error: %@", [NSString stringWithFormat:@"Error de conexión: %@", [error description]]);
}

- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    [connection release];

    JSONDecoder* decoder = [[JSONDecoder alloc] init];
    NSDictionary *listDictionary = [decoder objectWithData: datos];

    NSArray* people =[listDictionary objectForKey:@"menugroups"];

    for (NSDictionary *person in people) {
        NSDictionary *personDetails = [person objectForKey:@"menugroup"];
        NSLog(@"Id: %@", [personDetails objectForKey:@"id"]);
        NSLog(@"Nombre: %@", [personDetails objectForKey:@"name"]);
    }

    [datos release];
}
}
```

Figura III - 75 Código del Consumo del Servicio Web en iOS

Los datos obtenidos los presentamos por consola.

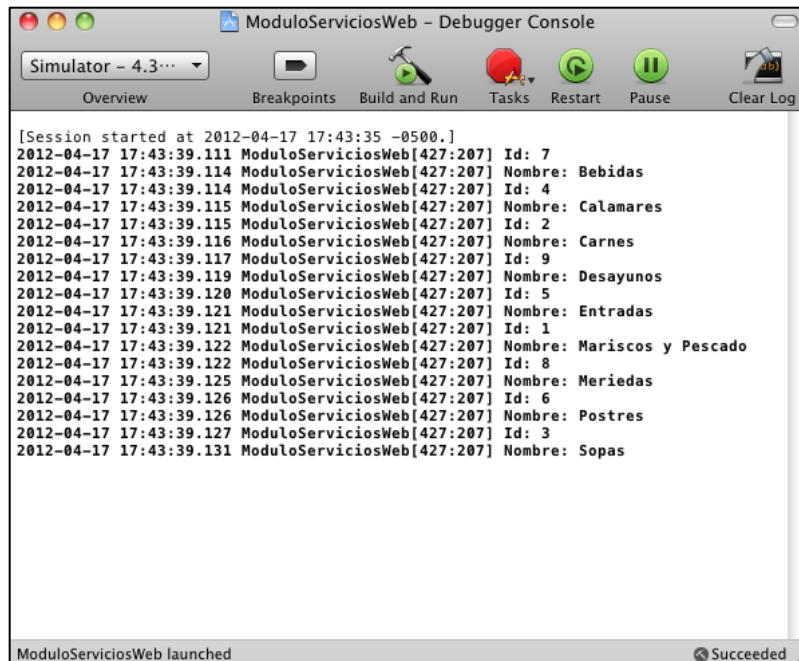
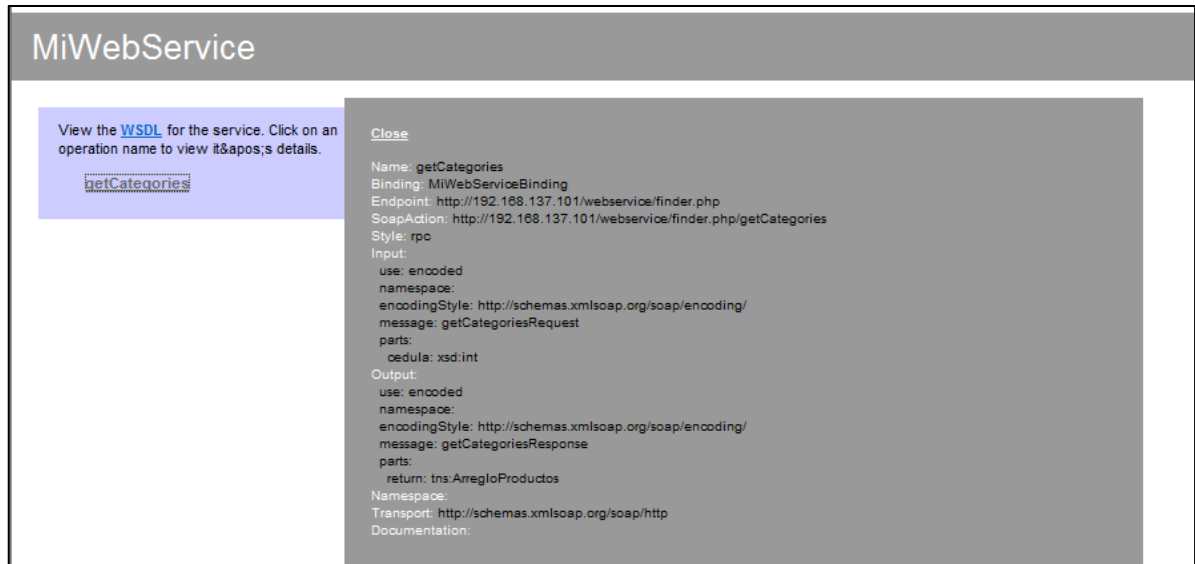


Figura III - 76 Resultado del Consumo del Servicio Web

Acceso a Servicios Web mediante SOAP

Para realizar el intercambio de datos mediante SOAP se utiliza el formato XML. Utilizaremos el servicio web publicado en el sitio web para la administración del restaurante que nos devuelve las categorías de platos en formato XML; el método publicado en el servicio web se llama get-categories.



The screenshot displays a web service interface titled "MiWebService". On the left, there is a blue box with the text: "View the [WSDL](#) for the service. Click on an operation name to view it's details." Below this, the operation name "getCategories" is listed. On the right, a "Close" button is visible above a detailed description of the "getCategories" operation. The details include:

- Name: getCategories
- Binding: MiWebServiceBinding
- Endpoint: http://192.168.137.101/webservice/finder.php
- SoapAction: http://192.168.137.101/webservice/finder.php/getCategories
- Style: rpc
- Input:
 - use: encoded
 - namespace:
 - encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
 - message: getCategoriesRequest
 - parts:
 - cedula: xsd:int
- Output:
 - use: encoded
 - namespace:
 - encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
 - message: getCategoriesResponse
 - parts:
 - return: tns:ArregloProductos
- Namespace:
- Transport: http://schemas.xmlsoap.org/soap/http
- Documentation:

Figura III - 77 Servicio Web con SOAP get-categories

Llamamos al servicio web y lo consumimos de la siguiente manera


```
- (void)viewDidLoad {
    [super viewDidLoad];
    NSString *soapMsg = @"<?xml version='1.0' encoding='utf-8'?>"
        "<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'"
        "xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>"
        "<soap:Body>"
        "<getCategories xmlns='http://192.168.137.101/webservice/'>"
        "<cedula>1</cedula>"
        "</getCategories>"
        "</soap:Body>"
        "</soap:Envelope>";
    NSURL *url = [NSURL URLWithString:@"http://192.168.137.101/webservice/finder.php"];
    NSString *msgLength = [NSString stringWithFormat:@"%d", [soapMsg length]];
    NSMutableURLRequest *req = [NSMutableURLRequest requestWithURL:url];

    [req addValue:@"text/xml; charset=utf-8" forHTTPHeaderField:@"Content-Type"];
    [req addValue:@"http://192.168.137.101/webservice/finder.php/getCategories" forHTTPHeaderField:@"SOAPAction"];
    [req addValue:msgLength forHTTPHeaderField:@"Content-Length"];
    [req setHTTPMethod:@"POST"];
    [req setHTTPBody:[soapMsg dataUsingEncoding:NSUTF8StringEncoding]];
    conn = [[NSURLConnection alloc] initWithRequest:req delegate:self];
    if (conn) {
        webData = [[NSMutableData data] retain];
    }
}

-(void) connection:(NSURLConnection *) connection didReceiveResponse:(NSURLResponse *) response {
    [webData setLength: 0];
}

-(void) connection:(NSURLConnection *) connection didReceiveData:(NSData *) data {
    [webData appendData:data];
}

-(void) connection:(NSURLConnection *) connection didFailWithError:(NSError *) error {
    [webData release];
    [connection release];
}

-(void) connectionDidFinishLoading:(NSURLConnection *) connection {
    NSString *theXML = [[NSString alloc] initWithBytes:[webData mutableBytes] length:[webData length]
        encoding:NSUTF8StringEncoding];
    NSLog(@"DATA: %@", theXML);
    [theXML release];
    [connection release];
    [webData release];
}
}
```

Figura III - 78 Consumo del Servicio Web con SOAP

Los datos obtenidos del consumo del servicio web son mostrados mediante la ventana de consola.

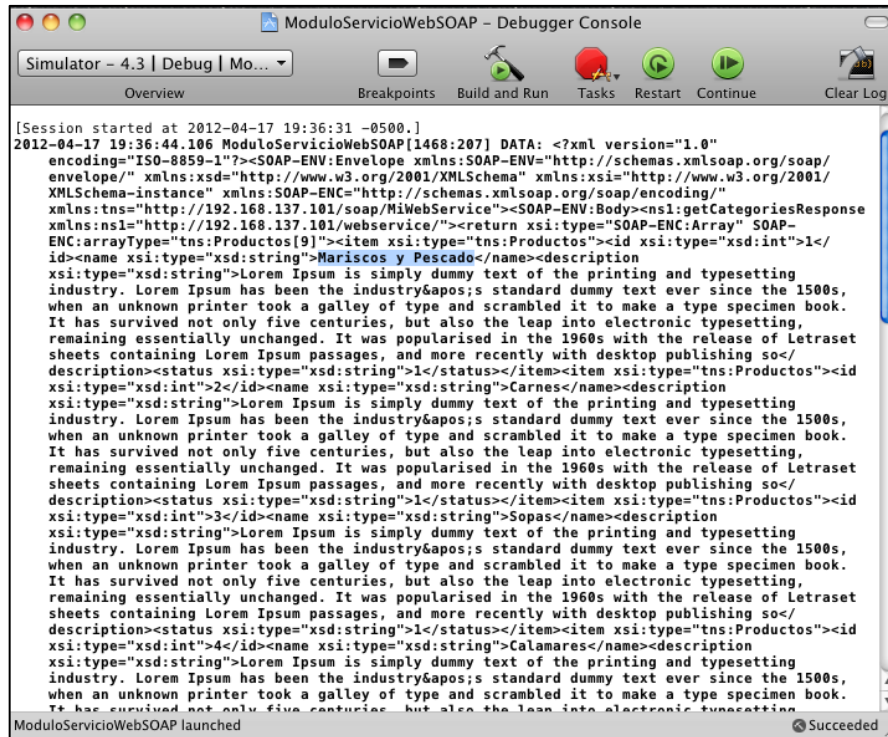


Figura III - 79 Resultado del Consumo del Servicio Web con SOAP

3.6.2.4 Módulos de Prueba de los Parámetros Entorno de Desarrollo, Compatibilidad entre Versiones y Facilidades de Despliegue.

3.6.2.4.1 Módulo 3 en Android

Para la construcción del módulo se va a utilizar el entorno de desarrollo Eclipse, el SDK de Android 4 y los emuladores que se encuentran incorporados con el SDK.

Los pasos para el desarrollo del módulo se describen a continuación:

Capacidad de ayuda inteligente

El IDE eclipse con la ayuda del SDK y el plugin Motodev Studio para Android tiene una completa ayuda inteligente la misma que nos permite vincularnos a la documentación del código (Ver Figura).

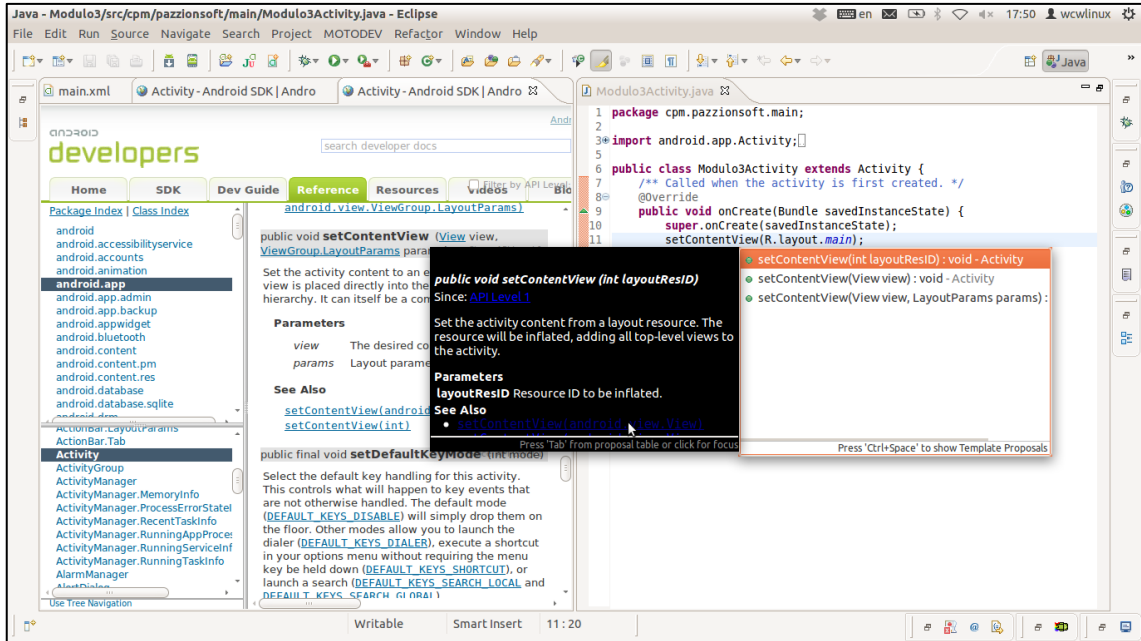


Figura III - 80 Ayuda Inteligente en Eclipse y Documentación de Android

Capacidad de Code Snippets

El IDE Eclipse con el SDK traen una característica fundamental que son los code snippets que son piezas o partes de código ya desarrollados como podemos visualizar en la siguiente figura:

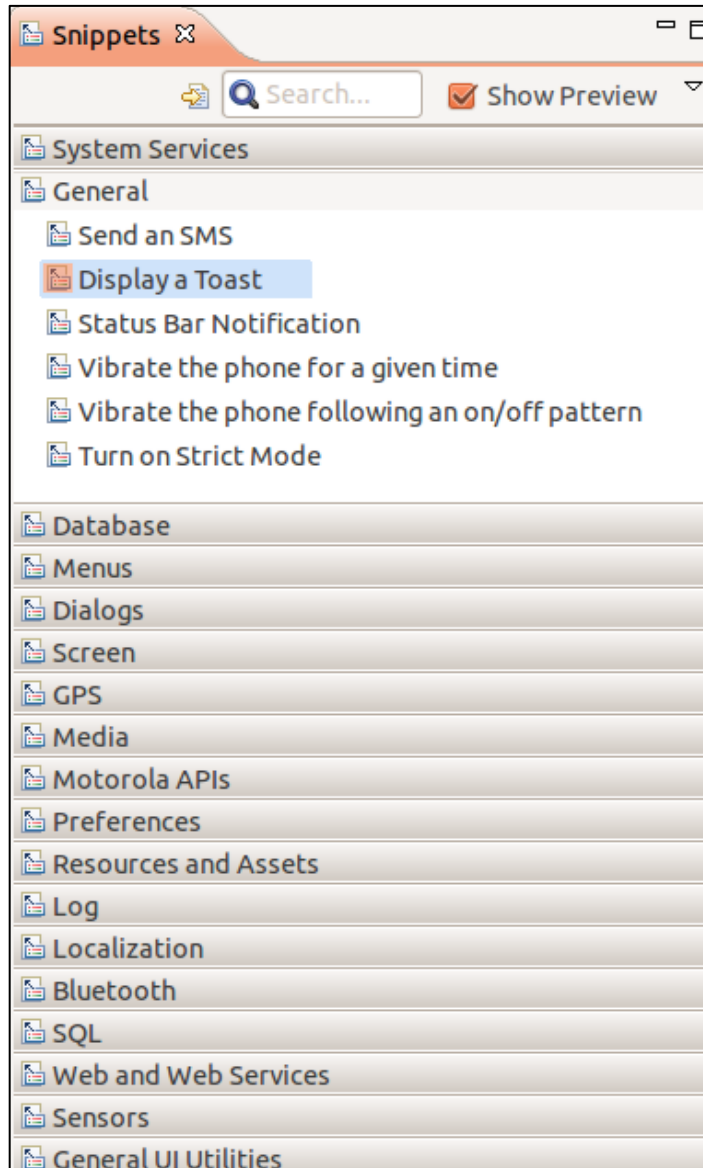


Figura III - 81 Conjunto de Code Snippets incorporados con el SDK

El SDK proporciona una variedad de código ya implementado como podemos ver a continuación.

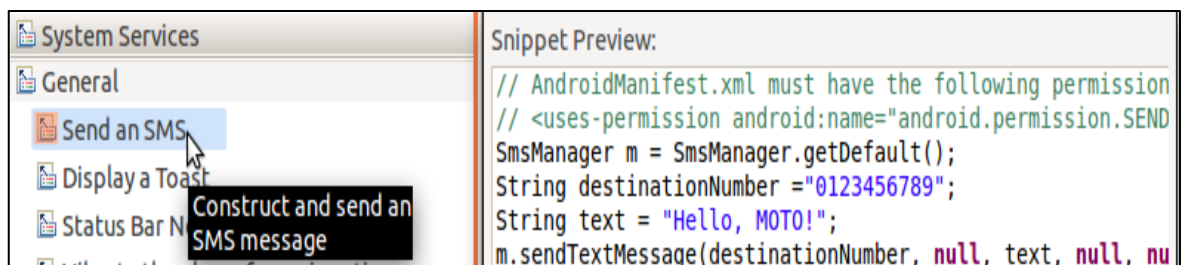


Figura III - 82 Ejemplo de Código Implementado mediante el SDK

El IDE eclipse proporciona algunas estructuras de código ya implementadas como podemos ver a continuación.

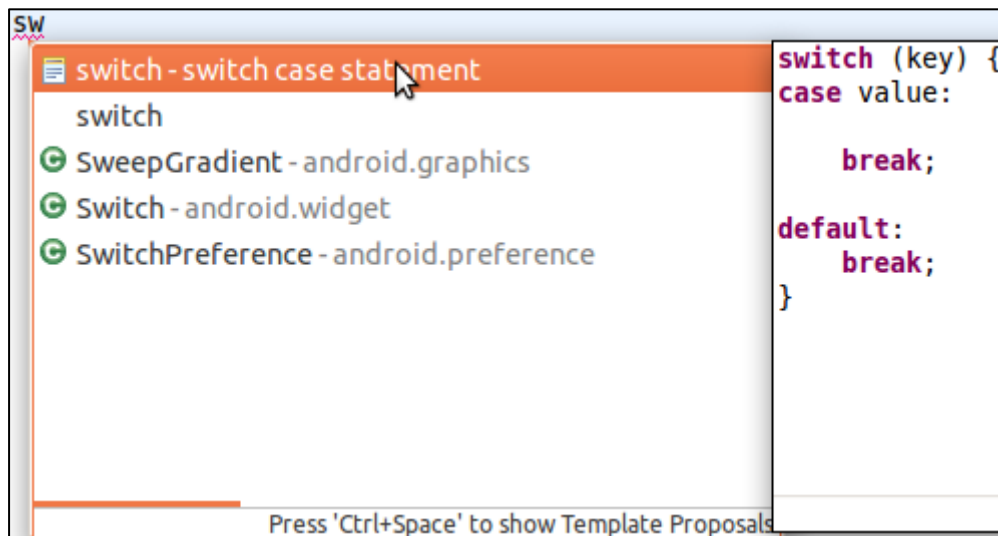


Figura III - 83 Ejemplo de Código Implementado mediante Eclipse

Facilidad de Depuración de aplicaciones

El plugin Motodev Studio para Android incluye una potente herramienta llamada Dalvik Debug Monitor, que será de gran ayuda para testear las aplicaciones, esta herramienta permite hacer un seguimiento de los puertos utilizados, ver mensajes de log, información acerca de los procesos en ejecución, gestión de memoria, así como otras muchísimas opciones de depuración.

A continuación observamos la ventana de depuración de código, en el cual se observa a detalle todas las opciones de depuración.

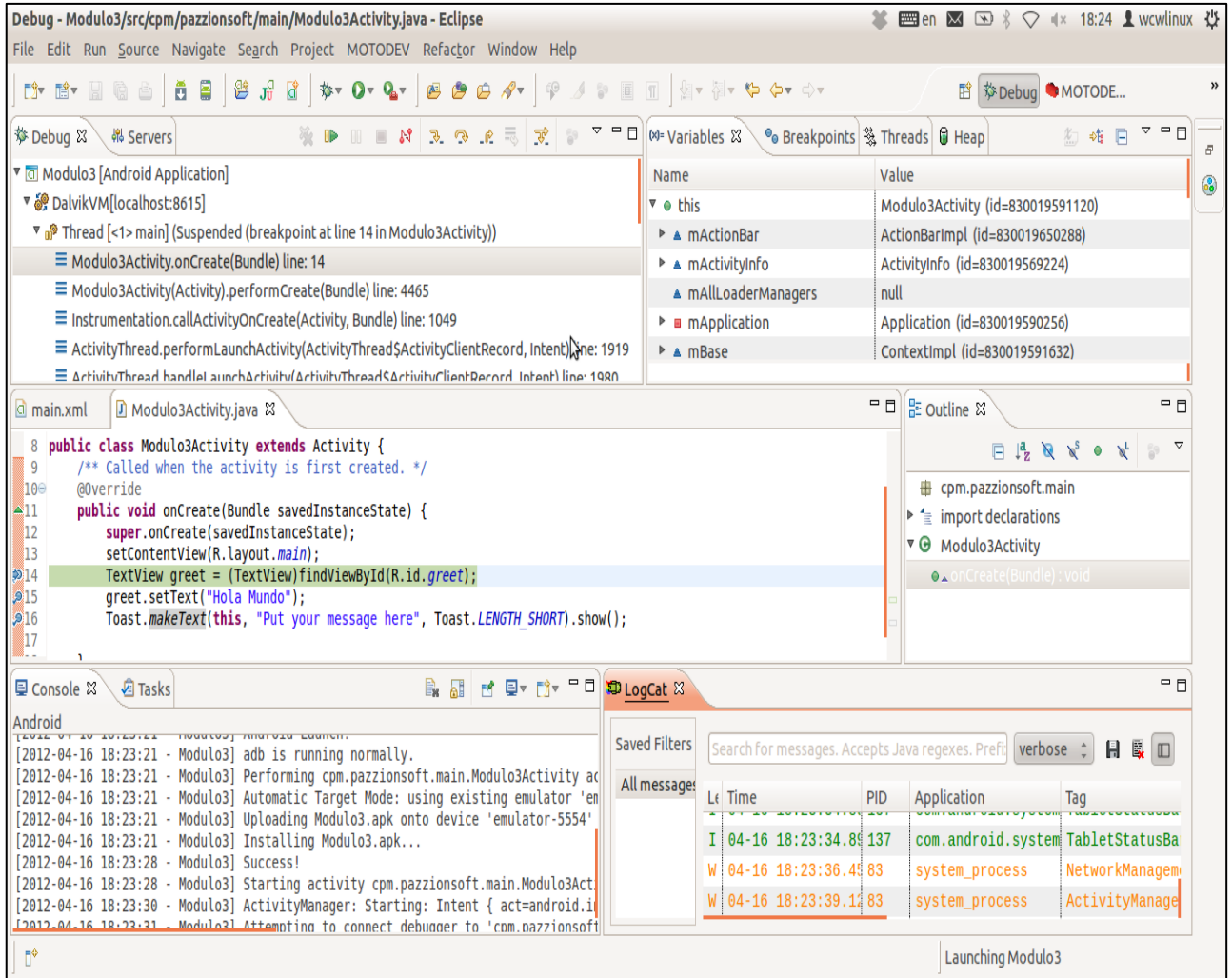


Figura III - 84 Ventana de Depuración de Código en Eclipse

Capacidad del Diseñador GUI Visual

El SDK proporciona varios elementos para el diseño de una interfaz visual así como una ventana de propiedades de cada elemento de la interfaz gráfica de usuario.

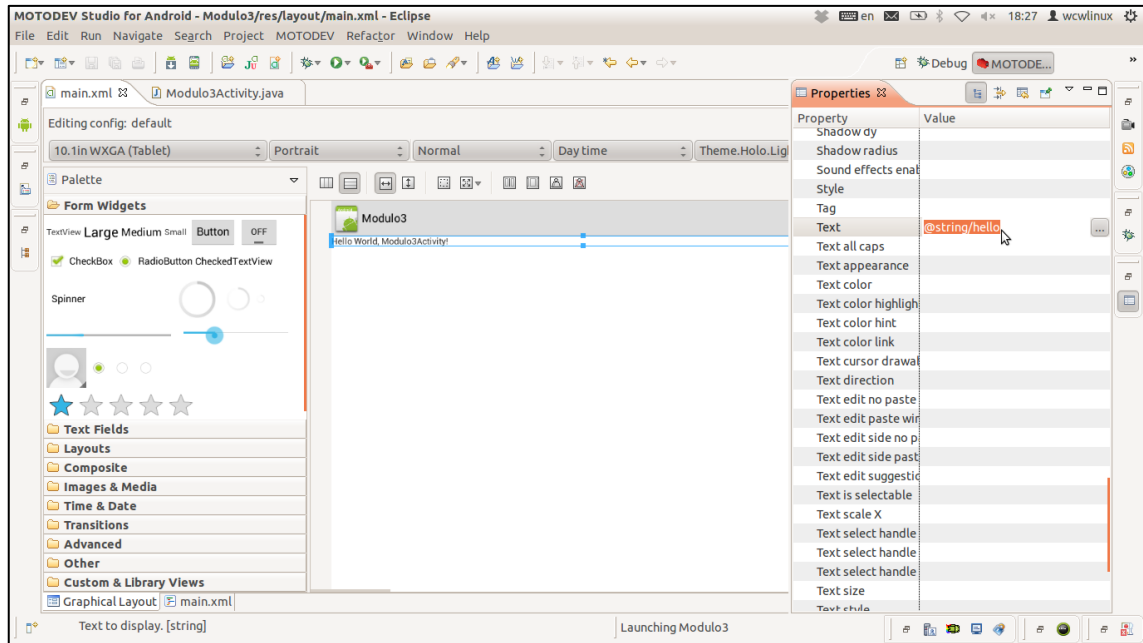


Figura III - 85 Elementos de Diseño de la Interfaz Gráfica para Android

Se puede arrastrar y soltar los elementos desde la paleta hacia el panel de diseño.

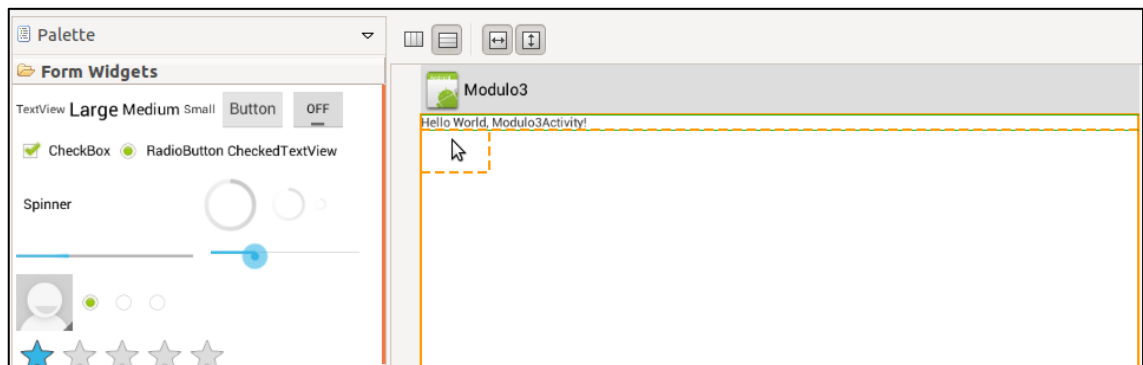


Figura III - 86 Elementos desde la Paleta de Componentes hacia el Panel

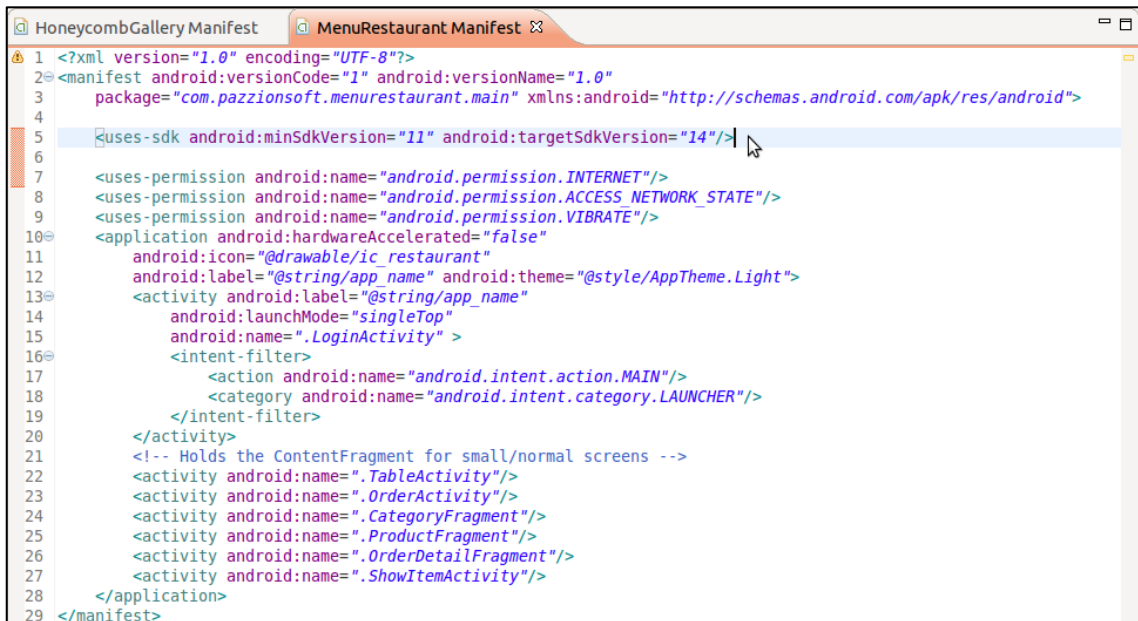
Capacidad de manejo de Versiones del SDK

Existen dos maneras de asignar las versiones del SDK.

La primera es asignando estos valores mediante un archivo xml como se muestra en la Figura.

El atributo **minSdkVersion** es obligatorio y es un número entero que representa el nivel mínimo del API requerido para que la aplicación funcione y este número debe ser menor o igual al sistema Android.

El atributo **targetSdkVersion** es un número que representa la versión en la que se desarrolla para el sistema Android.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <manifest android:versionCode="1" android:versionName="1.0"
3     package="com.pazzionsoft.menurestaurant.main" xmlns:android="http://schemas.android.com/apk/res/android">
4
5     <uses-sdk android:minSdkVersion="11" android:targetSdkVersion="14"/>
6
7     <uses-permission android:name="android.permission.INTERNET"/>
8     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
9     <uses-permission android:name="android.permission.VIBRATE"/>
10    <application android:hardwareAccelerated="false"
11        android:icon="@drawable/ic_restaurant"
12        android:label="@string/app_name" android:theme="@style/AppTheme.Light">
13        <activity android:label="@string/app_name"
14            android:launchMode="singleTop"
15            android:name=".LoginActivity" >
16            <intent-filter>
17                <action android:name="android.intent.action.MAIN"/>
18                <category android:name="android.intent.category.LAUNCHER"/>
19            </intent-filter>
20        </activity>
21        <!-- Holds the ContentFragment for small/normal screens -->
22        <activity android:name=".TableActivity"/>
23        <activity android:name=".OrderActivity"/>
24        <activity android:name=".CategoryFragment"/>
25        <activity android:name=".ProductFragment"/>
26        <activity android:name=".OrderDetailFragment"/>
27        <activity android:name=".ShowItemActivity"/>
28    </application>
29 </manifest>
```

Figura III - 87 Configuración de la Versión del SDK mediante XML

La segunda es definir estos valores mediante una ventana como se muestra en la Figura.

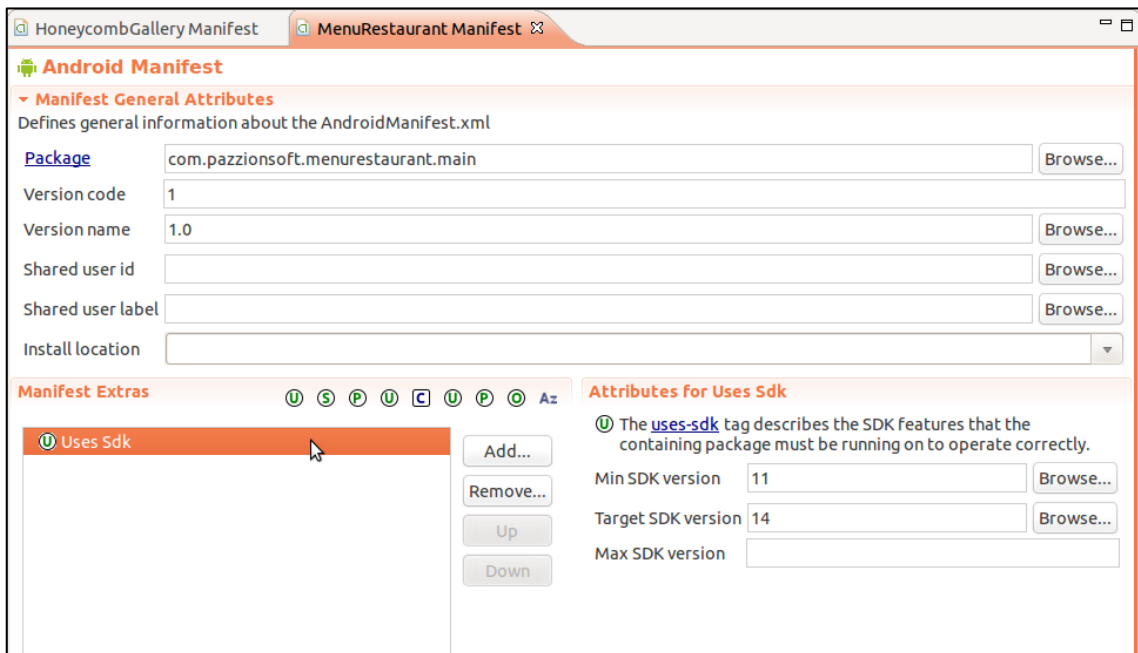


Figura III - 88 Configuración de la Versión del SDK mediante Interfaz Gráfica

Capacidad de compatibilidad de la aplicación para dispositivos de diferentes dimensiones de pantalla

Con la versión 4 de Android se puede desarrollar aplicaciones que sean compatibles tanto para smartphones como para tablets (opcional para el desarrollador) de diferentes resoluciones de pantalla como se puede observar la aplicación de ejemplo del SDK llamada HoneyCombGallery como se muestra en las siguientes figuras.

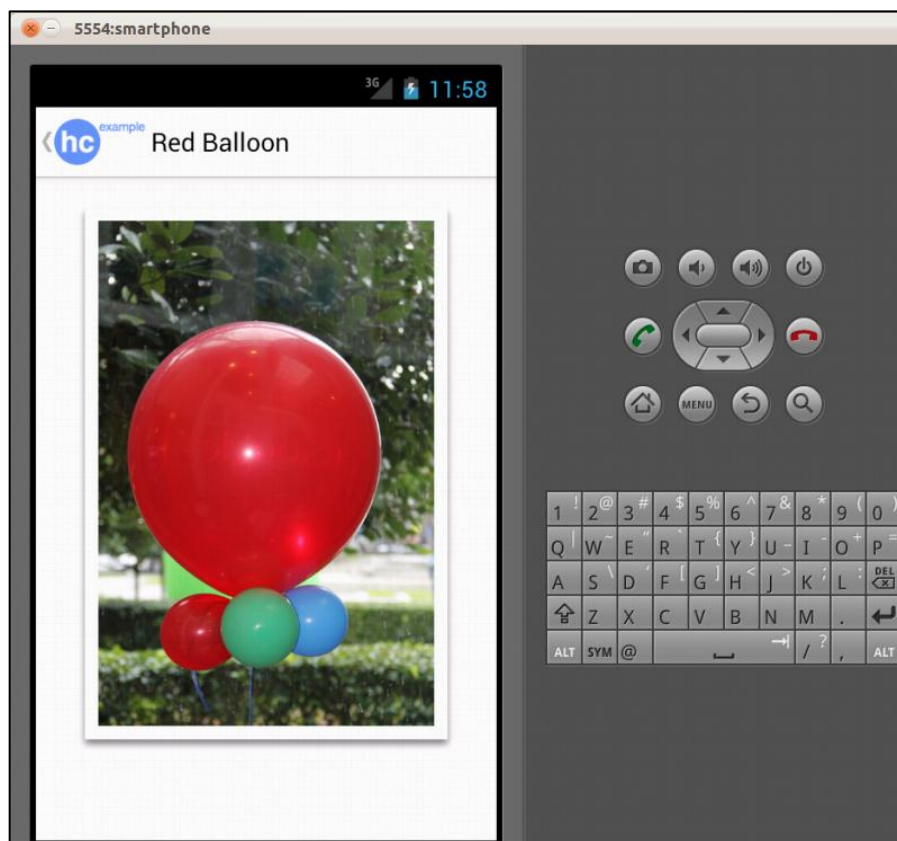


Figura III - 89 Aplicación HoneyCombGallery en Smartphone

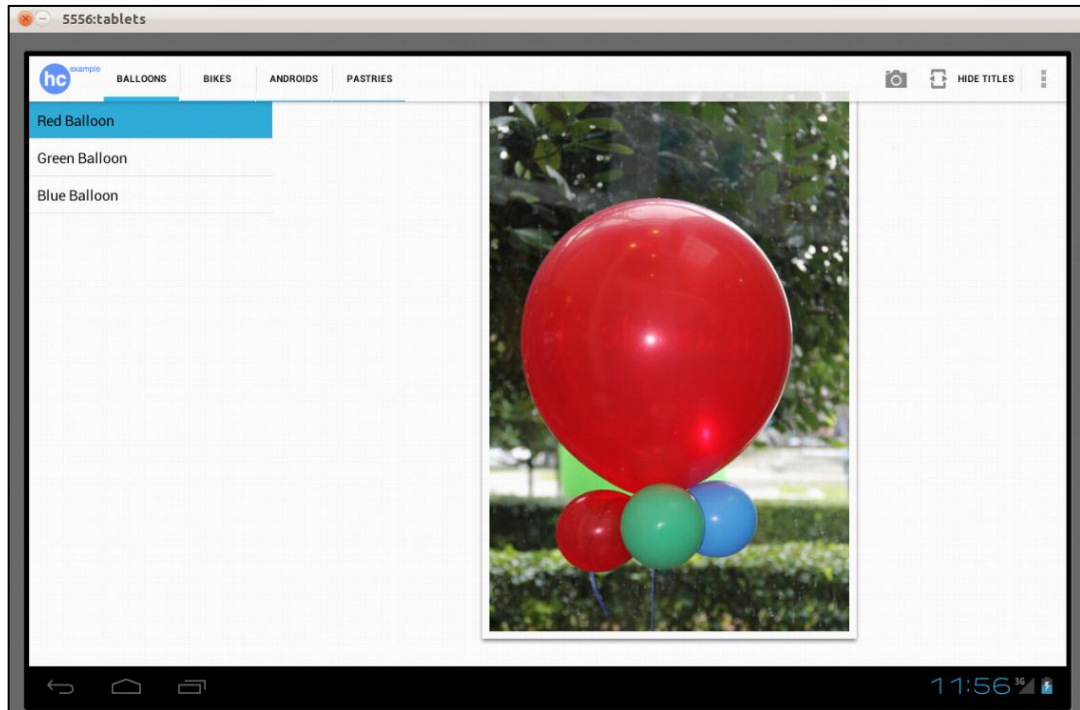


Figura III - 90 Aplicación HoneyCombGallery en Tablet

Disponibilidad de Emuladores

En el SDK de Android, se pueden crear emuladores de acuerdo a la versión del API y al tamaño de pantalla mediante una herramienta llamada Android Virtual Device Manager desde eclipse.

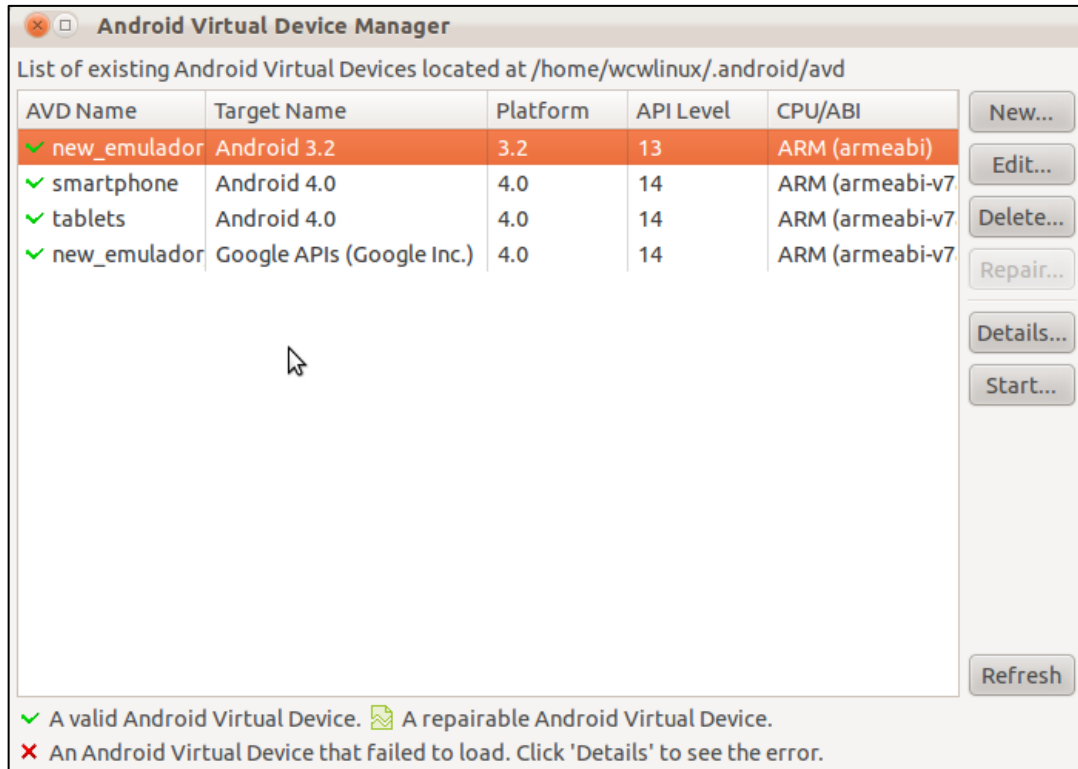


Figura III - 91 Ventana para la administración de Dispositivos Virtuales

Para crear una nueva Dispositivo Android Virtual llenamos los siguientes valores y damos click en Create ADV.

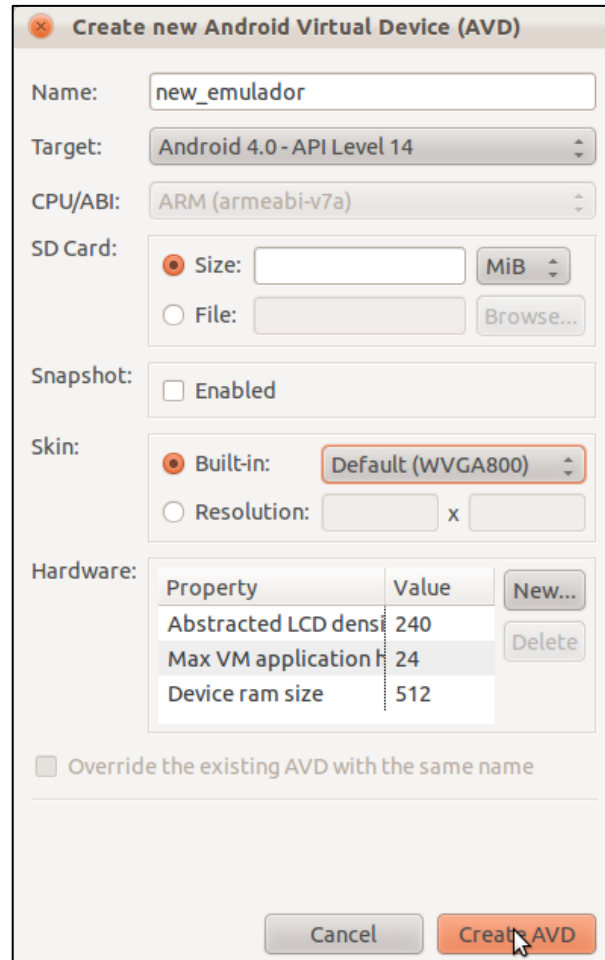


Figura III - 92 Ventana de Creación de un Nuevo Dispositivo Virtual

Facilidad para desplegar aplicaciones a un Dispositivo Físico

Para probar e instalar la aplicación en un dispositivo físico existen dos opciones:

La primera opción es ejecutar la aplicación en el dispositivo físico desde Eclipse:

Habilitamos el modo depuración en el dispositivo (Ajustes > Aplicaciones > Desarrollo > Depuración USB) y lo conectamos al ordenador mediante USB.

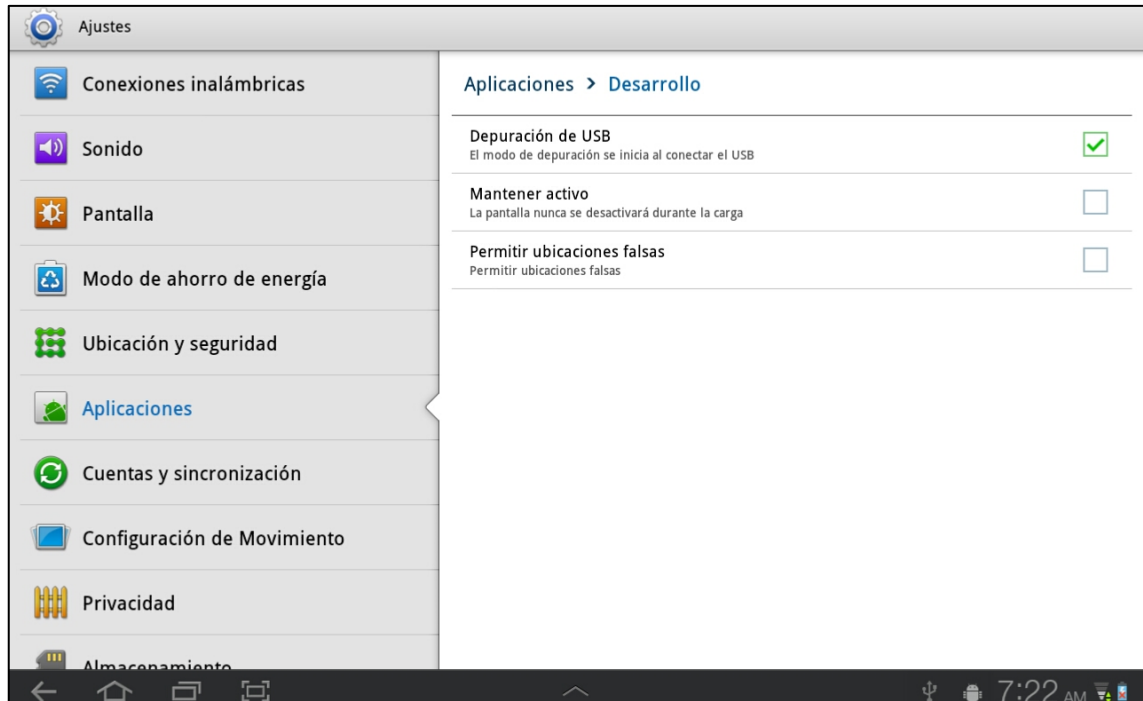


Figura III - 93 Habilitar el modo de depuración en el dispositivo

Eclipse lo reconocerá como un dispositivo en el que poder emular la aplicación y aparecerá en la lista de dispositivos cuando pulsemos Run. Solo será posible seleccionarlo en el caso de que el sistema operativo del terminal sea igual o superior al mínimo requerido por la aplicación.

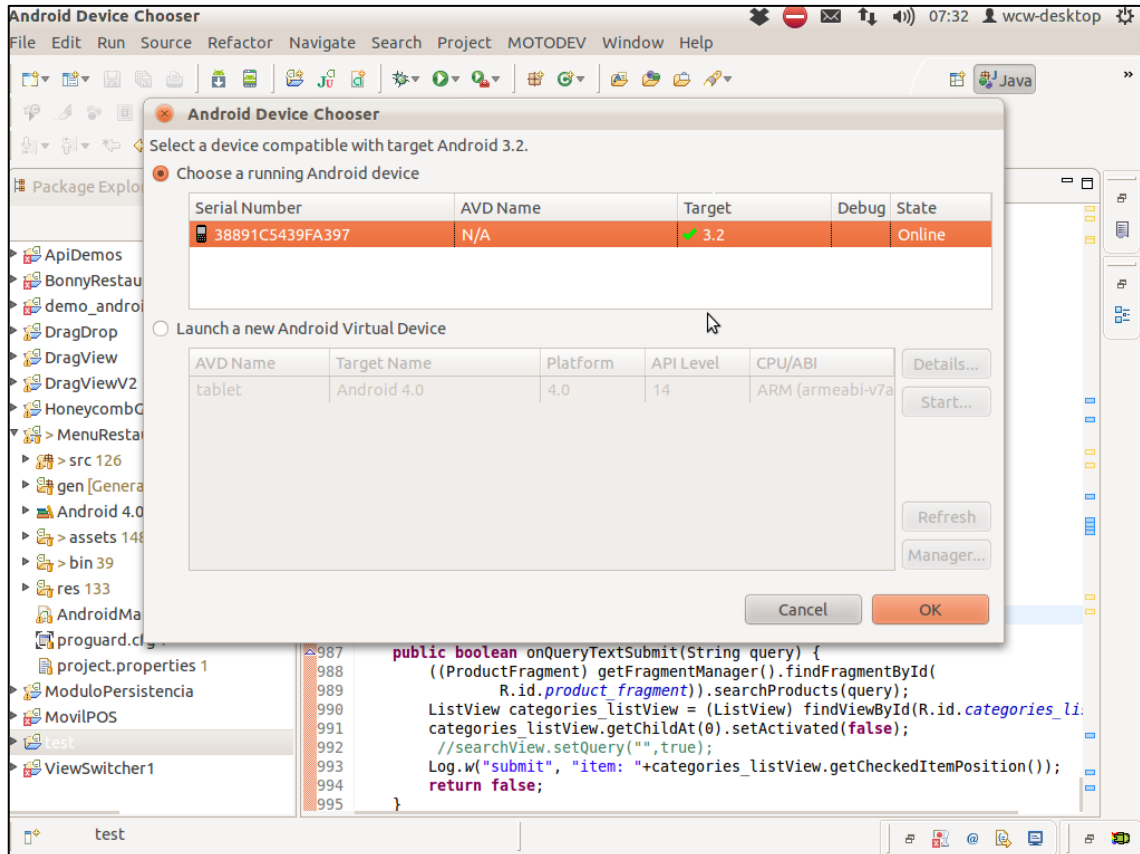


Figura III - 94 Lista de dispositivos físicos conectados

Si se selecciona, la aplicación se instalará en el dispositivo y se iniciará automáticamente, además quedará instalada aunque se desconecte del ordenador.

Si el dispositivo no puede conectarse al ordenador tenemos una segunda opción, podemos exportar nuestra aplicación desde Eclipse, creando un archivo .apk.

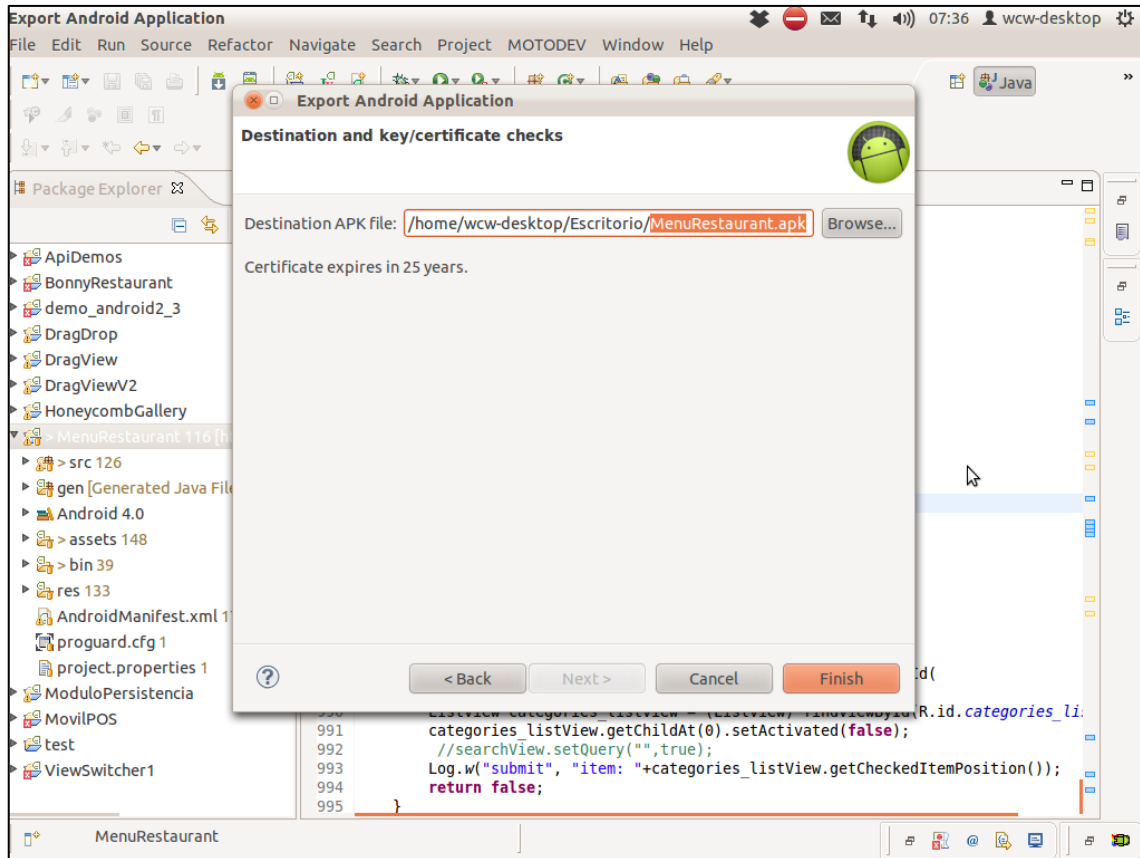


Figura III - 95 Exportación del .apk desde eclipse

El archivo podrá ser copiado en el dispositivo físico (descargándolo desde una web, copiándolo en la memoria) y luego, con un gestor de archivos en el terminal, se instalará manualmente. Por defecto, Android no deja que se instale aplicaciones de orígenes distintos a su Market . Esto puede habilitarse en el propio dispositivo en Ajustes > Aplicaciones > Fuentes Desconocidos.

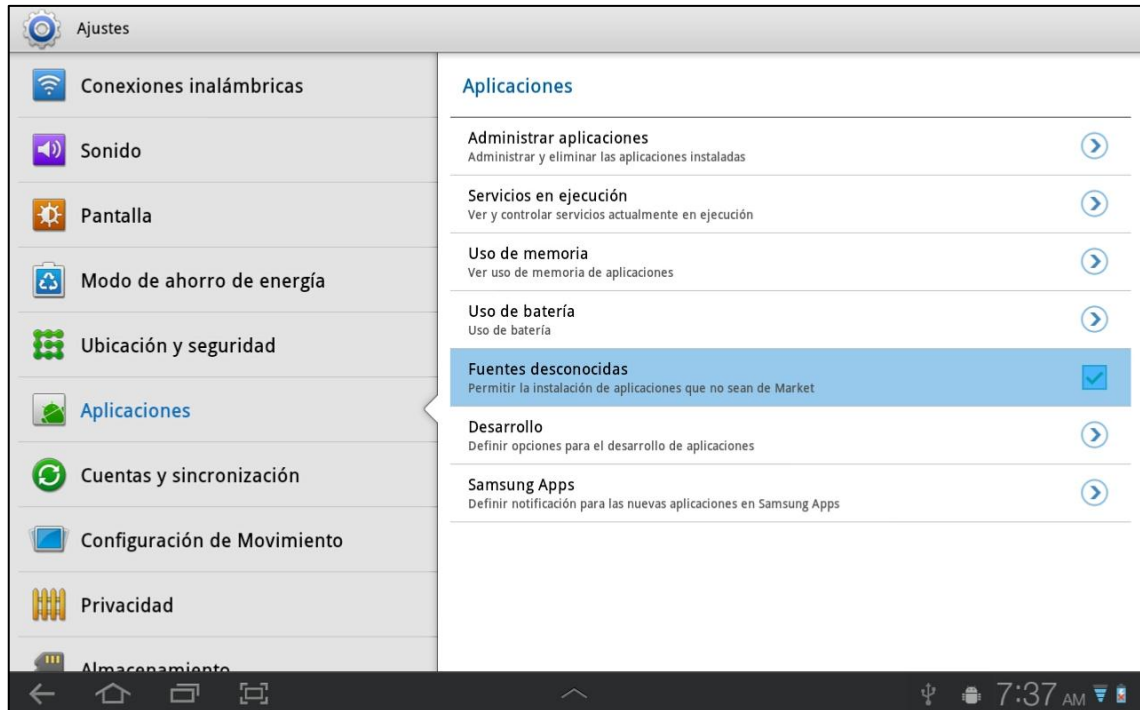


Figura III - 96 Habilitar Fuentes desconocidas para aplicaciones que no sean de Google Play

Fichero .apk

Un fichero de extensión .apk⁵¹ es un paquete de Android (Android Package), esto no es más que un archivo .zip con una estructura muy determinada que contiene los ficheros y las clases ya compiladas necesarias para ejecutar la aplicación. Para disponer de este archivo hay que exportarlo desde Eclipse (Export). Se mostrarán una serie de ventanas en las que habrá que determinar el modo en que se desea exportar el proyecto y el fichero clave con nuestros datos para firmar la aplicación.

3.6.2.4.2 Módulo 3 en iOS

Para la construcción del módulo se va a utilizar el entorno de desarrollo Xcode, el SDK de iOS 4.3 y los emuladores que se encuentran incorporados con el SDK.

⁵¹ <http://telekita.wordpress.com/2012/02/08/135/>

Los pasos para el desarrollo del módulo se describen a continuación:

Capacidad de ayuda inteligente

El IDE XCode con la ayuda del SDK para iOS tiene una completa ayuda inteligente la misma que nos permite vincularnos a la documentación del código (Ver Figura).

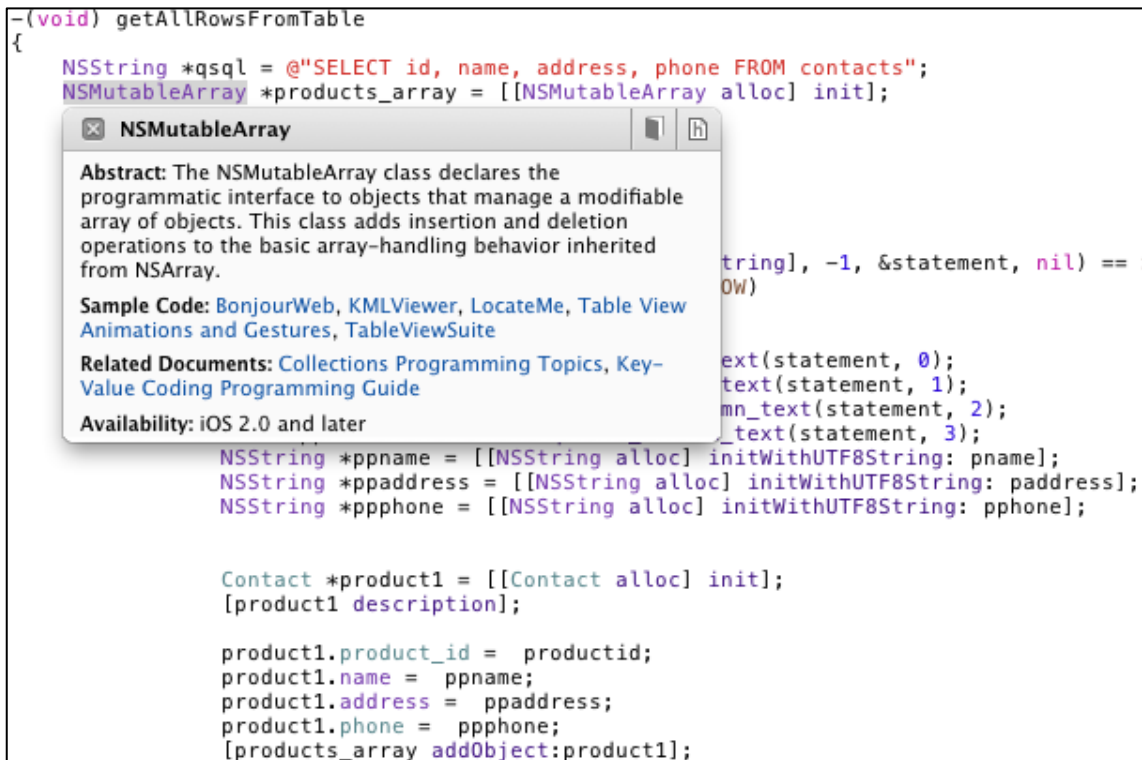


Figura III - 97 Documentación de un Tipo de Dato NSMutableArray

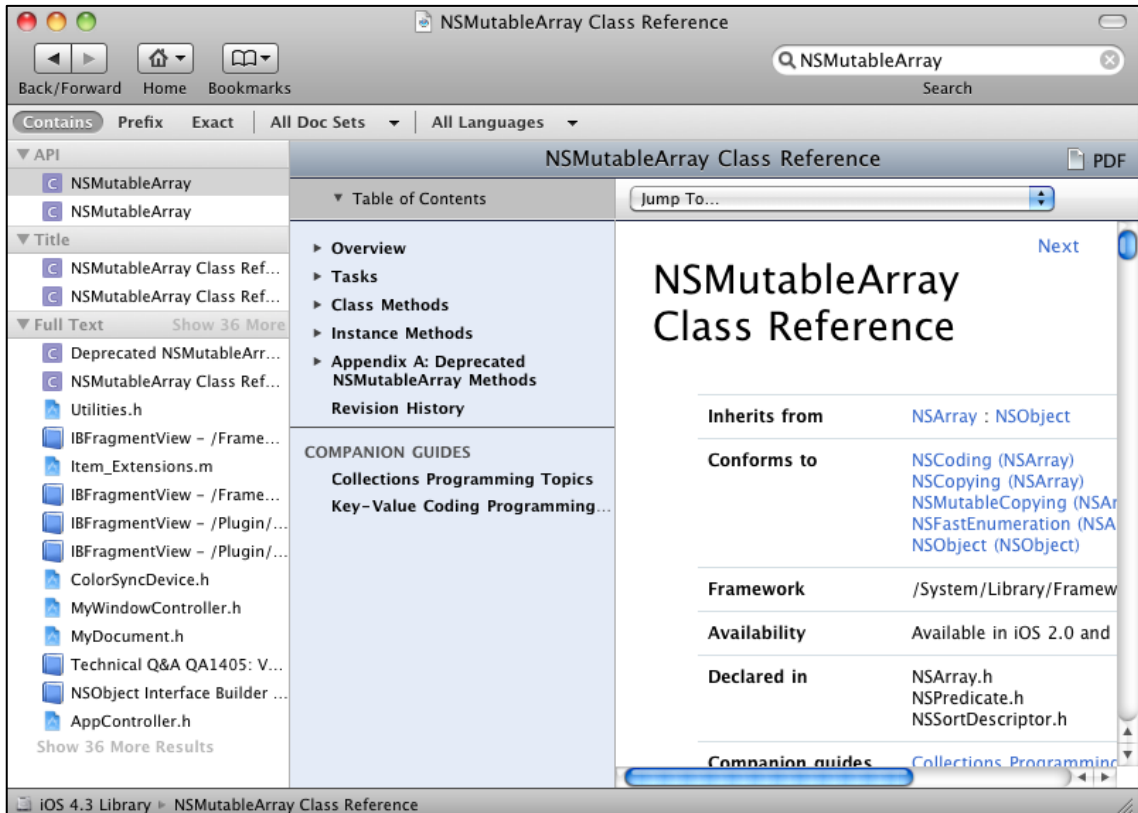


Figura III - 98 Detalle de la Documentación de un NSMutableArray

Capacidad de Code Snippets

En XCode contiene code snippets que son piezas o partes de código ya desarrollados de estructuras básicas fundamentales, donde con solamente escribir la parte del código se puede dar un enter y aceptar la estructura, como podemos visualizar en la siguiente figura:

```
switch (expression) {  
    case constant:  
        statements  
        break;  
    default:  
        break;  
}
```

Figura III - 99 Bloque de Código de un Switch

Facilidad de Depuración de aplicaciones

XCode incluye una herramienta para testear las aplicaciones, esta herramienta permite ver mensajes de log, información acerca de los procesos en ejecución, gestión de memoria, así como otras opciones de depuración.

A continuación observamos la ventana de depuración de código, en el cual se observa a detalle todas las opciones de depuración.

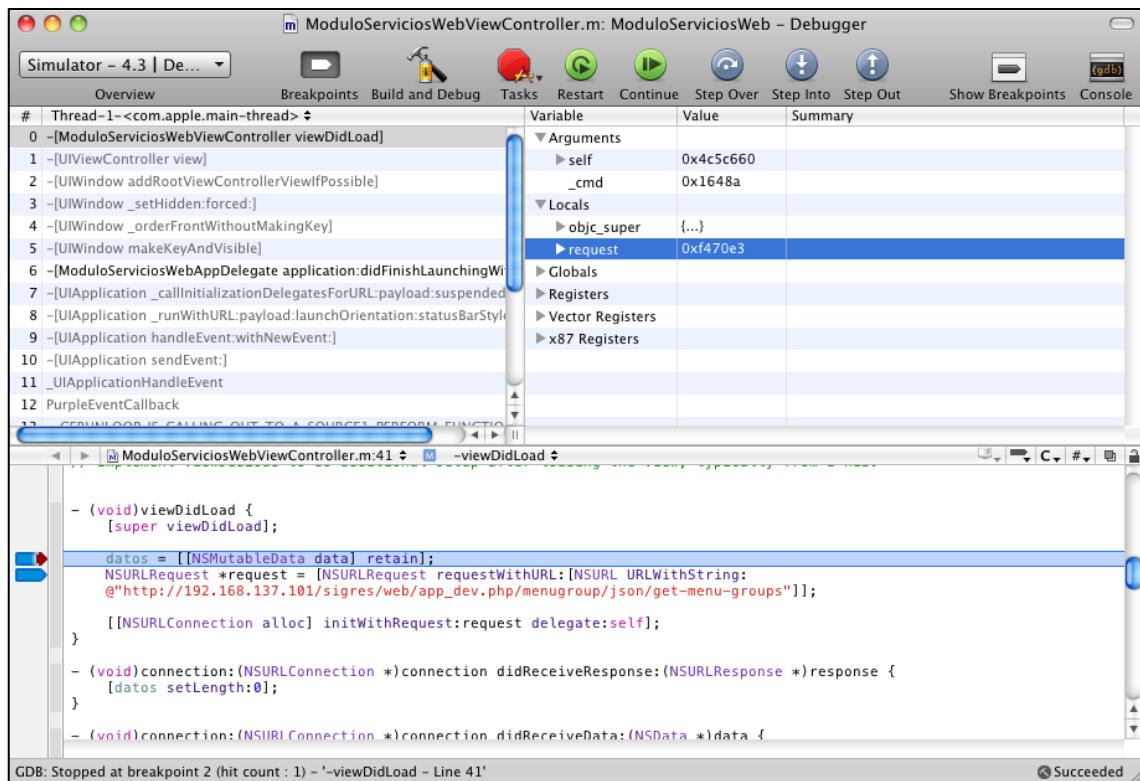


Figura III - 100 Ventana de Depuración de Código en XCode iOS

Capacidad del Diseñador GUI Visual

XCode proporciona varios elementos para el diseño de una interfaz visual así como una ventana de propiedades de cada elemento de la interfaz gráfica de usuario donde con sus distintas opciones se pueden personalizar cada elemento.

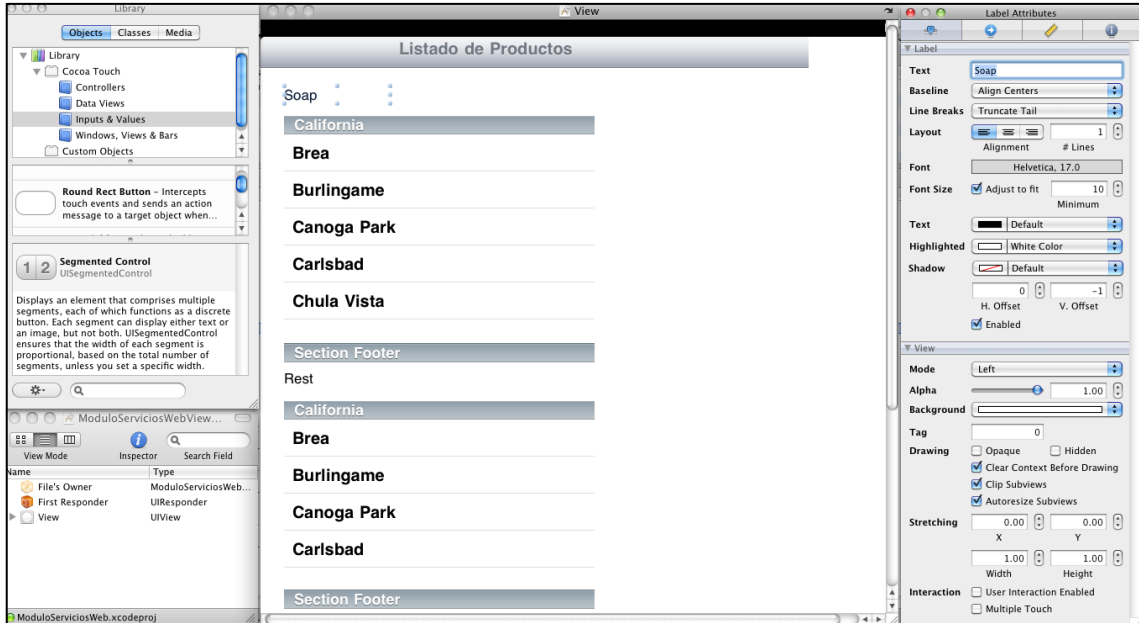


Figura III - 101 Diseñador de Interfaz Gráfica

Se puede arrastrar y soltar los elementos desde la paleta hacia el panel de diseño.

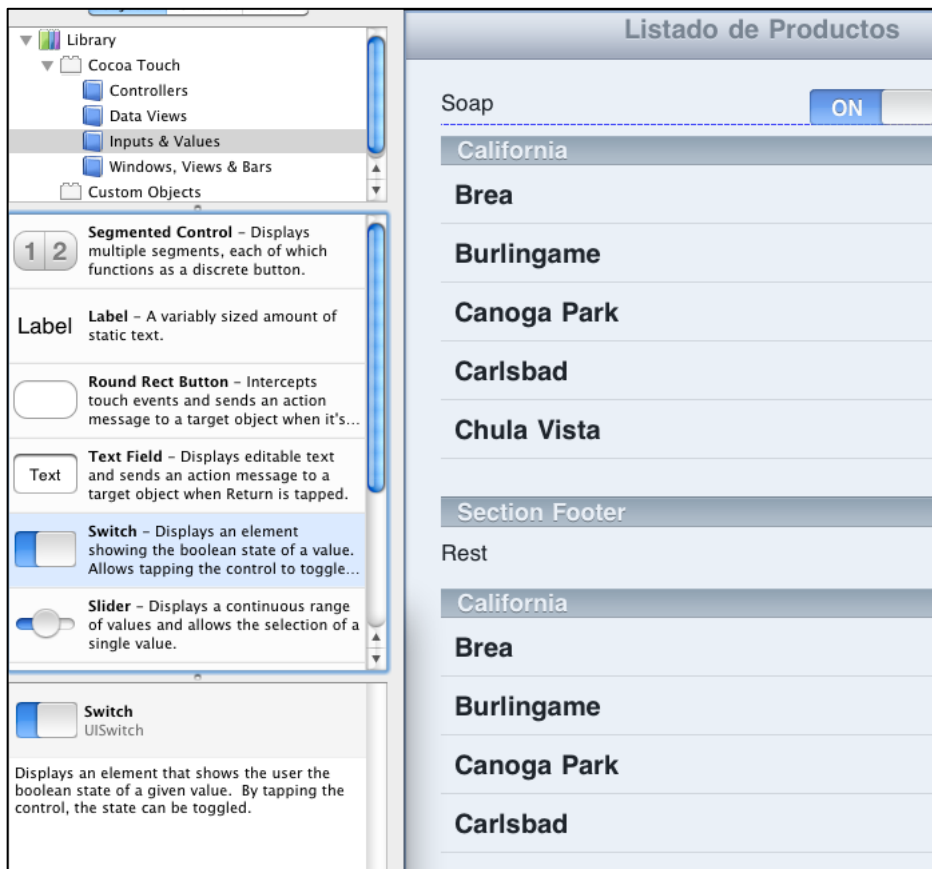


Figura III - 102 Listado de Elementos de Interfaz Gráfica

Capacidad de manejo de Versiones del SDK

Existen dos maneras de asignar las versiones del SDK.

Se realiza editando las configuraciones del proyecto desde XCode.

El atributo **Base SDK** es la versión del SDK que se utiliza cuando se compila la aplicación por lo tanto es la versión en la que fue desarrollada la aplicación iOS.

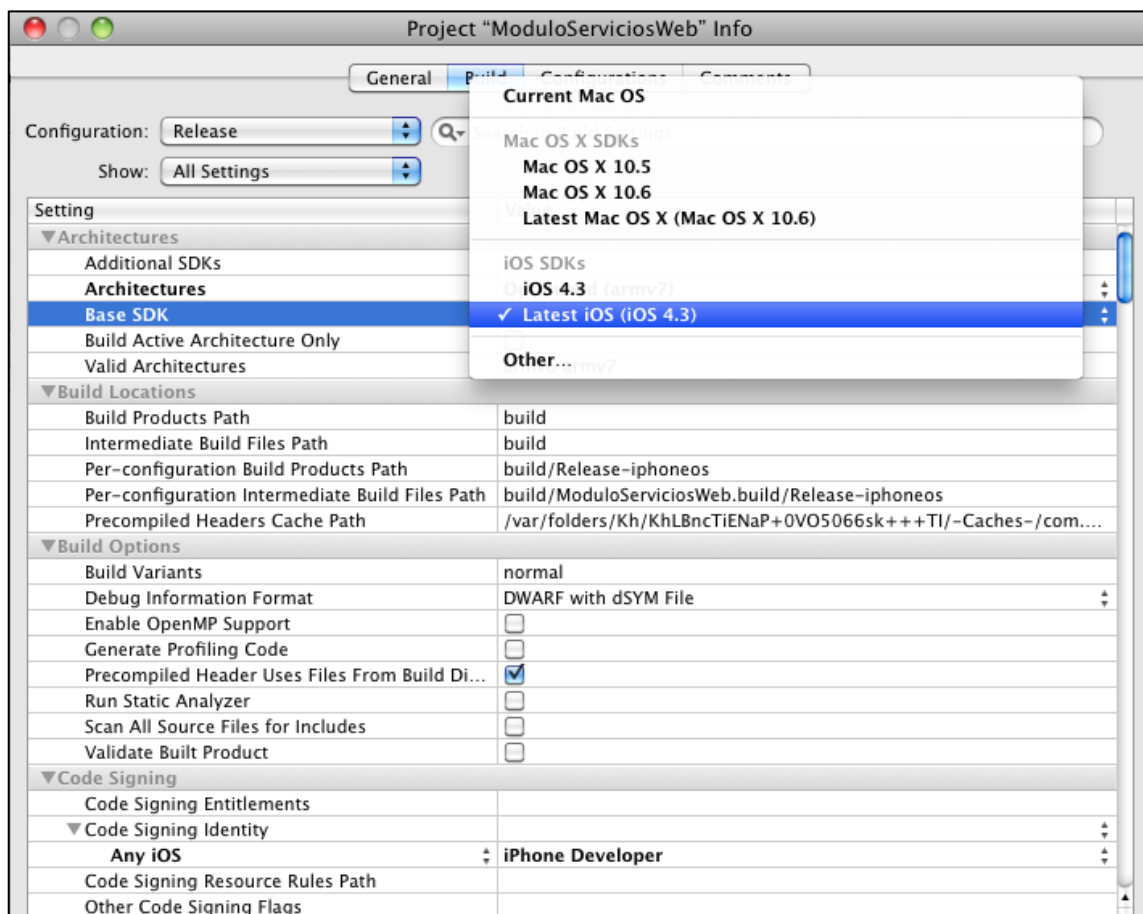


Figura III - 103 Configuración del Atributo Base SDK

El atributo **iOS Deployment Target** sirve para especificar que versión del SDK es el mínimo en que la aplicación se va a ejecutar.

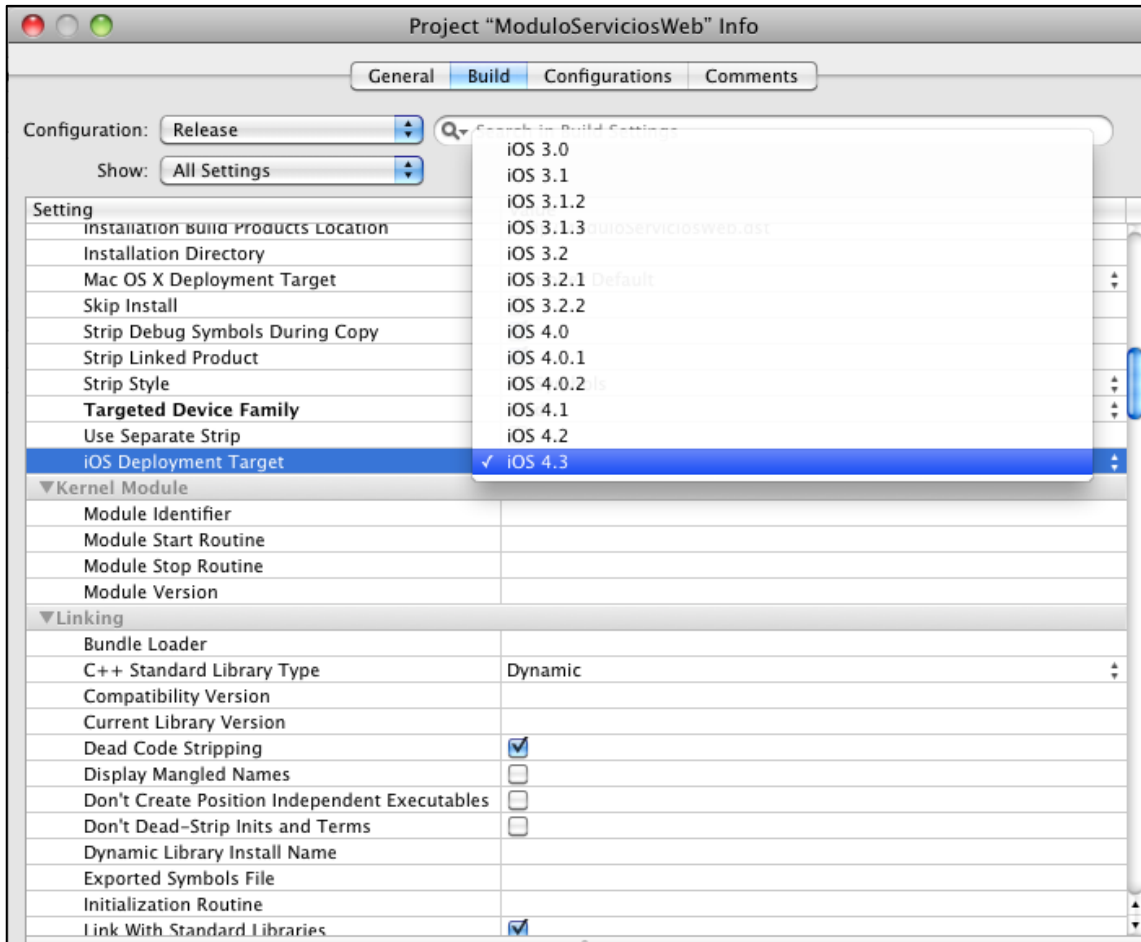


Figura III - 104 Configuración del Atributo iOS Deployment Target

Capacidad de compatibilidad de la aplicación para dispositivos de diferentes dimensiones de pantalla

Una aplicación desarrollada para el iPhone puede ejecutarse en el iPad ya que el iPad tiene mayor tamaño de pantalla que el iPhone pero una aplicación realizada para el iPad no se puede ejecutar en el iPhone.

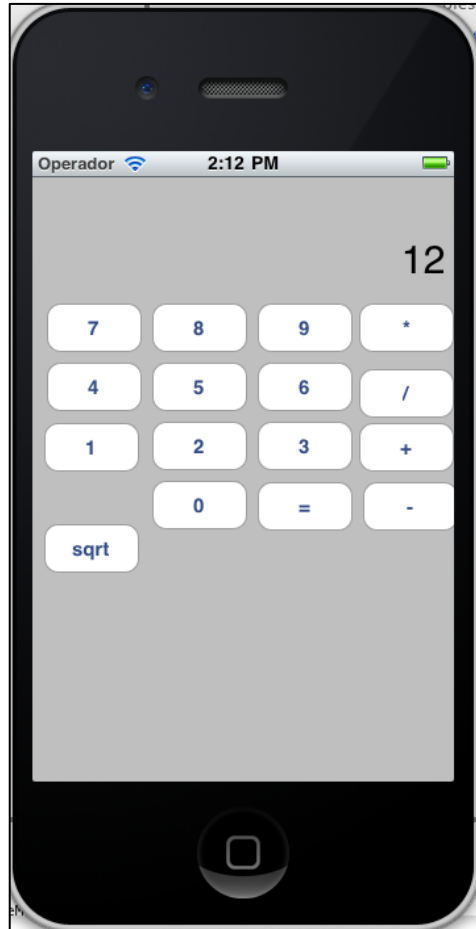


Figura III - 105 Aplicación desarrollada para iPhone

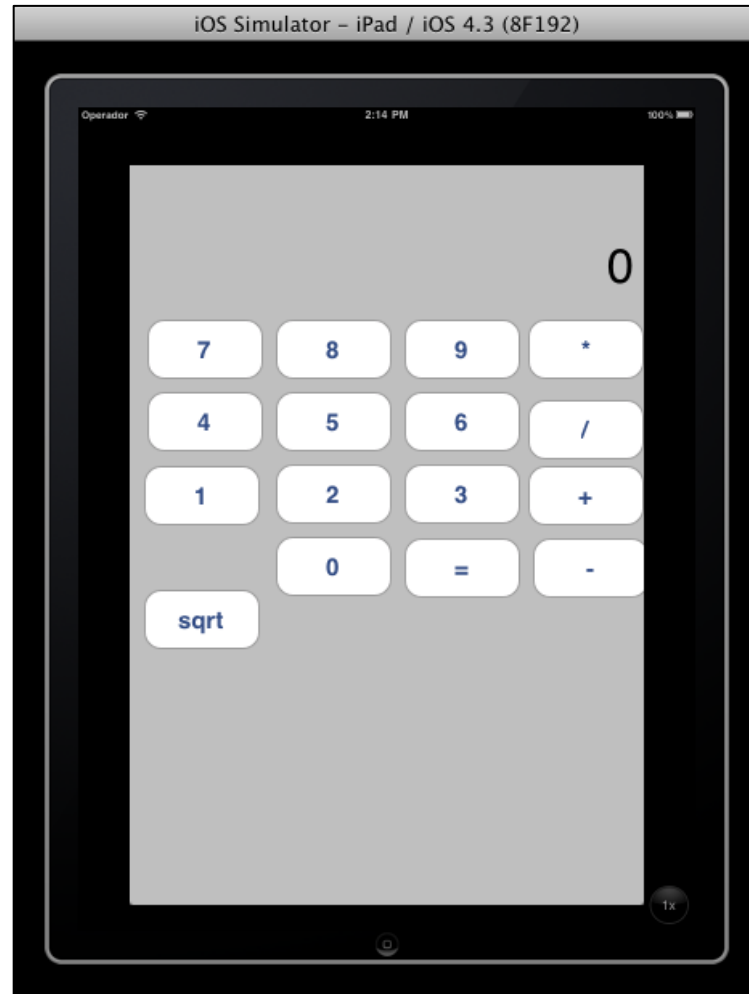


Figura III - 106 Aplicación desarrollada en iPhone ejecutada en el iPad

Disponibilidad de Emuladores

En el SDK de iOS, se pueden utilizar dos emuladores para realizar pruebas en las aplicaciones el uno sirve para ejecutar las aplicaciones para iPad y el otro para el iPhone, cada uno en sus diferentes versiones.



Figura III - 107 Emulador del iPad

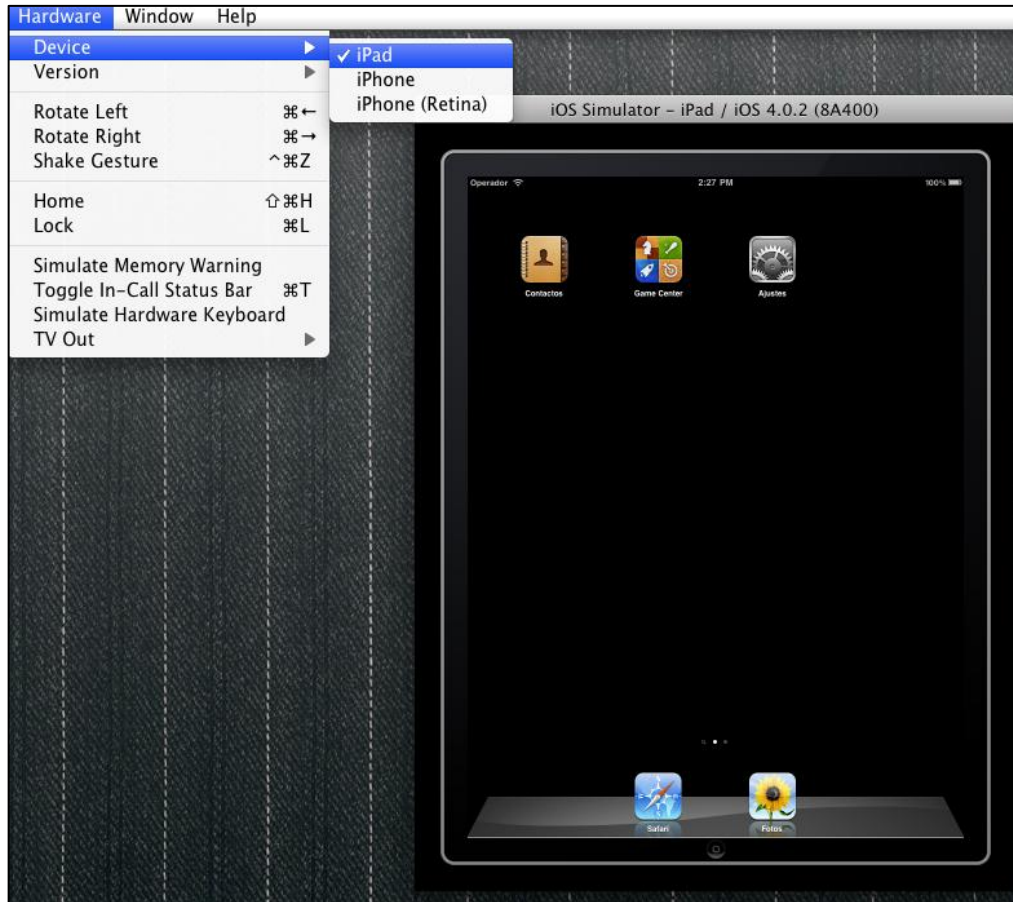


Figura III - 108 Emulador del iPad

Se debe seleccionar la versión en la que se va a ejecutar el emulador del iPad.

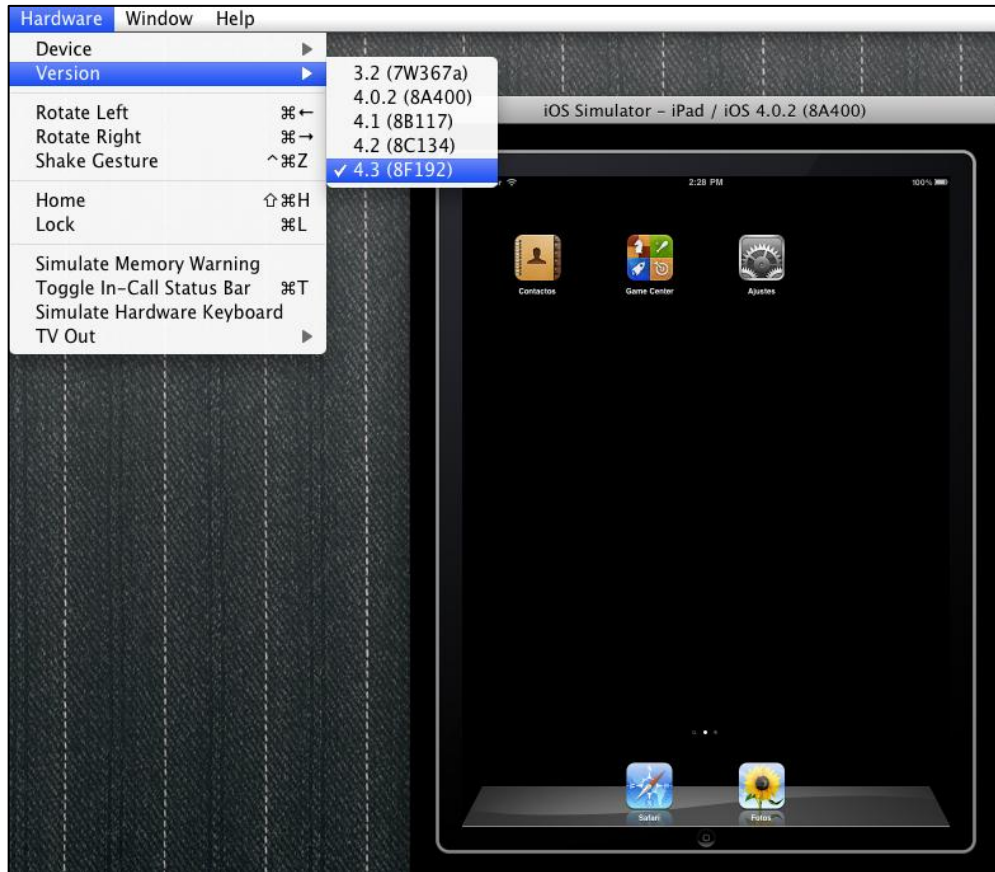


Figura III - 109 El nombre del proyecto es ModuloServiciosWeb

Facilidad para desplegar aplicaciones a un Dispositivo Físico

Para el despliegue de las aplicaciones hacia el dispositivo físico (con el fin de realizar pruebas) vamos a realizar los siguientes pasos:

1. Registrarse como cuenta pagada de Apple Developer. Sólo las cuentas de pago se puede implementar en dispositivos físicos.
2. Abrir Xcode 4 del Organizador, y seleccionar "Provisioning Profiles" en el menú de la izquierda:

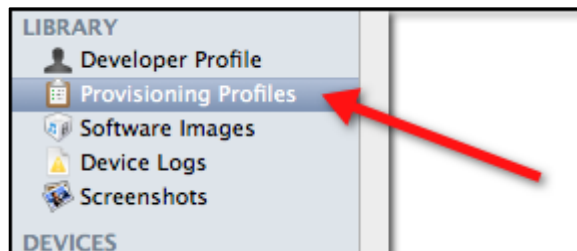


Figura III - 110 Ventana Developer Profile

3. En la parte inferior del lado derecho, "marcar" la casilla de verificación "Automatic Device Provisioning":



Figura III - 111 Ventana Automatic Device Provisioning

4. Conectar el dispositivo a la computadora. Seleccionar el dispositivo en el menú de la izquierda, y hacer clic en la opción "Use for Development" botón:



Figura III - 112 Ventana de Dispositivos

5. Se pedirá que inicie sesión en la cuenta de desarrollador de Apple. Introducir las credenciales aquí:

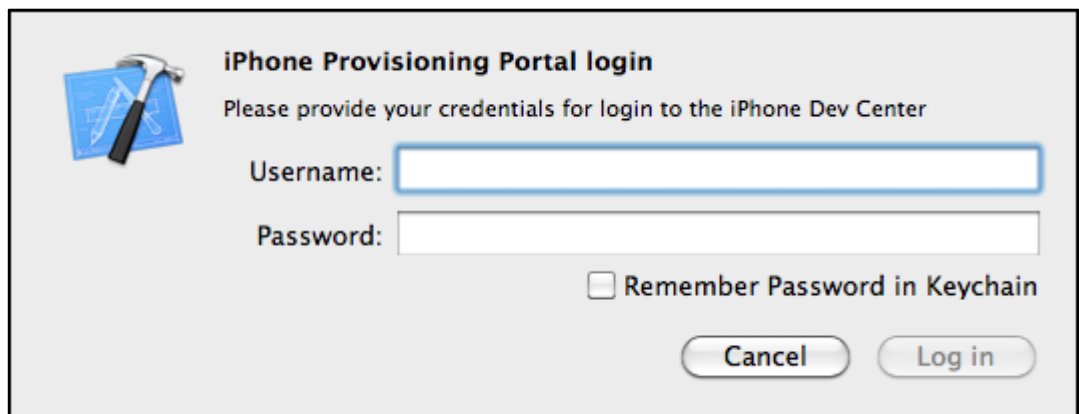


Figura III - 113 Ventana de Login de la Cuenta de Desarrollador

6. Xcode es probable que se le diga que no reconoce la versión de IOS. Deje que se recabará la información necesaria cuando se le solicite:

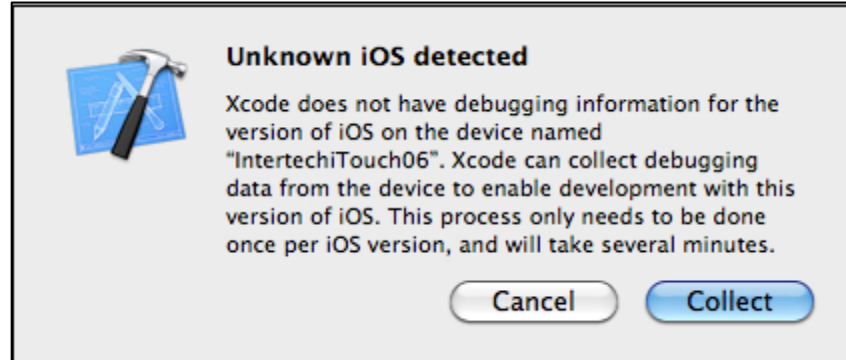


Figura III - 114 Ventana de Reconocimiento del Dispositivo

7. Al finalizar debe aparecer el dispositivo en la lista de la lista de la parte superior izquierda de "esquemas" en Xcode:

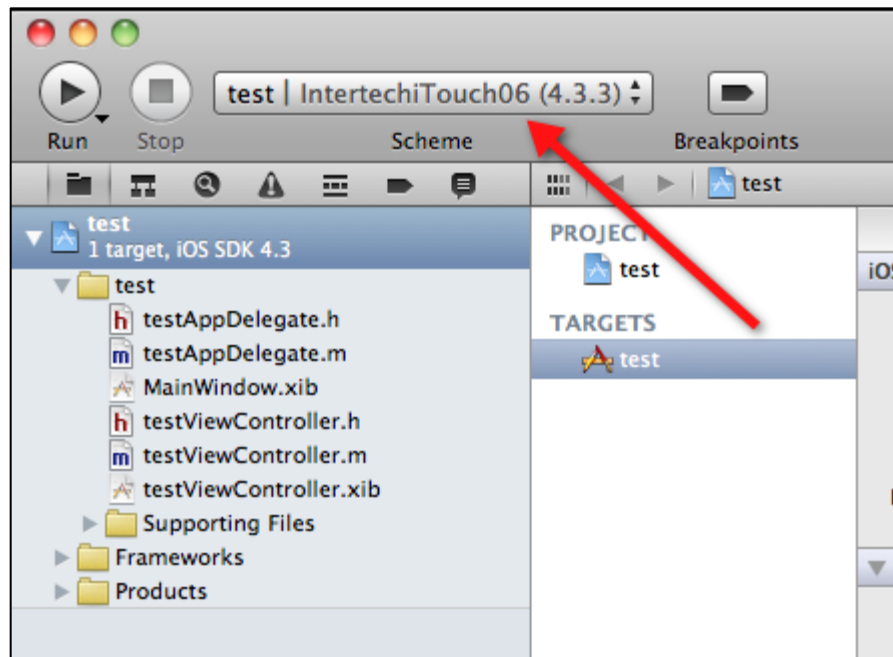


Figura III - 115 Ventana Esquemas en XCode

3.7 Análisis Comparativo

En esta sección se va a mostrar el análisis de las plataformas de desarrollo para dispositivos móviles inteligentes Android y iOS a manera de implementación de módulos en ambas plataformas, seguido de una interpretación y calificación del criterio evaluado por parte de los autores de la tesis.

Para obtener los resultados cuantitativos y cualitativos que permitan una selección sustentada de una de las dos plataformas de desarrollo para dispositivos móviles inteligentes analizados, se calificó cada uno de los parámetros de comparación, basados en la siguiente escala:

TABLA III - XII Escala de Valoraciones

Índice	Valor
EA	1
EB	2
EC	3
ED	4

TABLA III - XIII Escala de Valoraciones Cualitativa

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Ninguno	Pocos	Varios	Muchos
No satisfactorio	Poco satisfactorio	Satisfactorio	Muy satisfactorio
Malo	Regular	Bueno	Muy bueno
Inadecuado	Más o menos	Adecuado	Muy adecuado
Insuficiente	Parcial	Suficiente	Excelente
Deficiente	Poco eficiente	Eficiente	Muy eficiente
Ningún Avance	Cierto avance	Avance significativo	Objetivo Logrado

Fuente: Proporcionada por el asesor de la tesis

Dónde:

EA = Evaluación Mala

EB = Evaluación Regular

EC = Evaluación Bueno

ED = Evaluación Muy Bueno

TABLA III - XIV Pesos para los Parámetros

DESCRIPCIÓN	PESO
PV1	=[EA, EB, EC, ED]
PV2	=[EA, EB, EC, ED]
...	...
PVi	=[EA, EB, EC, ED]

Dónde:

PV1, PV2, ..., PVi = Peso para las variables en las dos tecnologías en el parámetro.

El valor de cada variable en su respectivo parámetro de interpretación para la tecnología incluye la siguiente fórmula:

$$V1 \geq 0 \quad V1 \leq PV1$$

$$V2 \geq 0 \quad V2 \leq PV2$$

...

$$Vi \geq 0 \quad Vi \leq PVi$$

Dónde:

Vi = Valor de cada variable en el parámetro.

La calificación definitiva de la herramienta en base a cada parámetro de comparación se obtiene sumando los puntajes obtenidos del análisis, utilizando las siguientes fórmulas:

$$\mathbf{Pan} = \sum (Vi)$$

$$P_{io} = \sum (V_j)$$

$$P_c = \sum (P_{Vi})$$

Dónde:

Pan = Puntaje acumulado por Android en el parámetro.

Pio = Puntaje acumulado por iOS en el parámetro.

Pc = Puntaje sobre el que se califica el parámetro.

TABLA III - XV Porcentajes de los Resultados

DESCRIPCIÓN	RESULTADO
Can	$= (P_{an} / P_c) * 100\%$
Cio	$= (P_{io} / P_c) * 100\%$

Dónde:

Can: Calificación que obtuvo la Plataforma Android en el parámetro

Cio: Calificación que obtuvo la Plataforma iOS en el parámetro

TABLA III - XVI Calificación de las Tecnologías

DESCRIPCIÓN	CALIFICACIÓN
Regular	$< 70\%$
Bueno	$\geq 70\% \text{ y } < 80\%$
Muy bueno	$\geq 80 \text{ y } < 95\%$
Excelente	$\geq 95\%$

3.7.1 Lenguaje de Programación

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XVII Escala de Valoraciones Cualitativa para el Parámetro Lenguajes de Programación

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Ninguno	Pocos	Varios	Muchos
Malo	Regular	Bueno	Muy bueno

Determinación de los Indicadores

TABLA III - XVIII Calificación de los Indicadores del Parámetro Lenguajes de Programación

INDICADORES	ANDROID	IOS
Cantidad de lenguajes de programación.	Pocos	Pocos
Curva de aprendizaje de los lenguajes de programación.	Muy Bueno	Regular

Fuente: Ver Módulo Teórico

Interpretación de Resultados

$$P_{an} = \sum (V_i)$$

$$P_{an} = \sum (2 + 4)$$

$$P_{an} = 6$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (2 + 2)$$

$$P_{io} = 4$$

$$P_c = \sum (P_{V_i})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (6 / 8) * 100\%$$

$$C_{an} = 75\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$Cio = (4 / 8) * 100\%$$

$$Cio = 50\%$$

Interpretación

- En Android ofrece dos lenguajes de programación que son: Java y C/C++. Y en iOS soporta dos lenguajes de programación que son: Objective C y C. Por lo que ambas plataformas ofrecen pocos lenguajes de programación.

No se encontró igualdad en el aprendizaje de los lenguajes de programación de Android y iOS, ya que Java tiene una curva de aprendizaje muy buena, porque tiene similitud con otros lenguajes como C++, C#, en el caso de iOS el lenguaje Objective C tiene una de aprendizaje regular por su sintaxis.

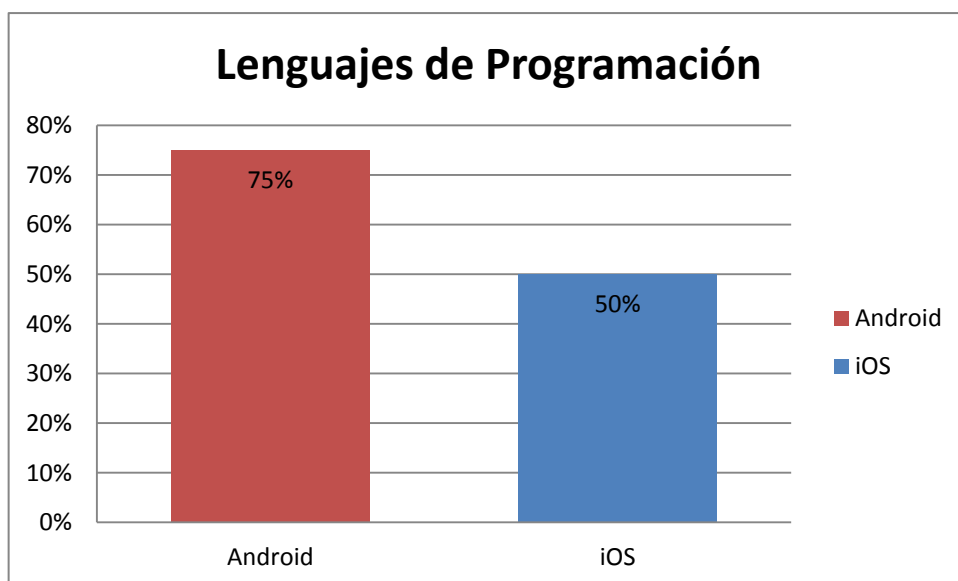


Figura III - 116 Resultado del Parámetro Lenguajes de Programación

3.7.2 Sistemas Operativos

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XIX Escala de Valoraciones Cualitativa para el Parámetro Sistemas Operativos

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Ninguno	Pocos	Varios	Muchos

Determinación de los Indicadores

TABLA III - XX Calificación de las Variables del Parámetro Sistemas Operativos

INDICADORES	ANDROID	IOS
Cantidad de sistemas operativos soportados	Muchos	Pocos
Cantidad de sistemas operativos gratuitos	Pocos	Ninguno

Fuente: Ver Módulo Teórico

Interpretación de Resultados

$$P_{an} = \sum (V_i)$$

$$P_{an} = \sum (4 + 2)$$

$$P_{an} = 6$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (2 + 0)$$

$$P_{io} = 2$$

$$P_c = \sum (P_{V_i})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (6 / 8) * 100\%$$

$$C_{an} = 75\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (2 / 8) * 100\%$$

$$C_{io} = 25\%$$

Interpretación

- Los Sistemas Operativos para la plataforma Android son Windows, Linux y Mac. El Sistema Operativo para la plataforma iOS es Mac OS X. Por lo que para la plataforma Android existen muchos Sistemas Operativos y para la plataforma iOS existen pocos Sistemas Operativos.

La plataforma Android utiliza como sistema operativo gratuito a Linux, mientras que la plataforma iOS no utiliza ningún Sistema Operativo Gratuito. Por lo que para Android existen pocos Sistemas Operativos Gratuitos mientras que en iOS no existe ninguno.

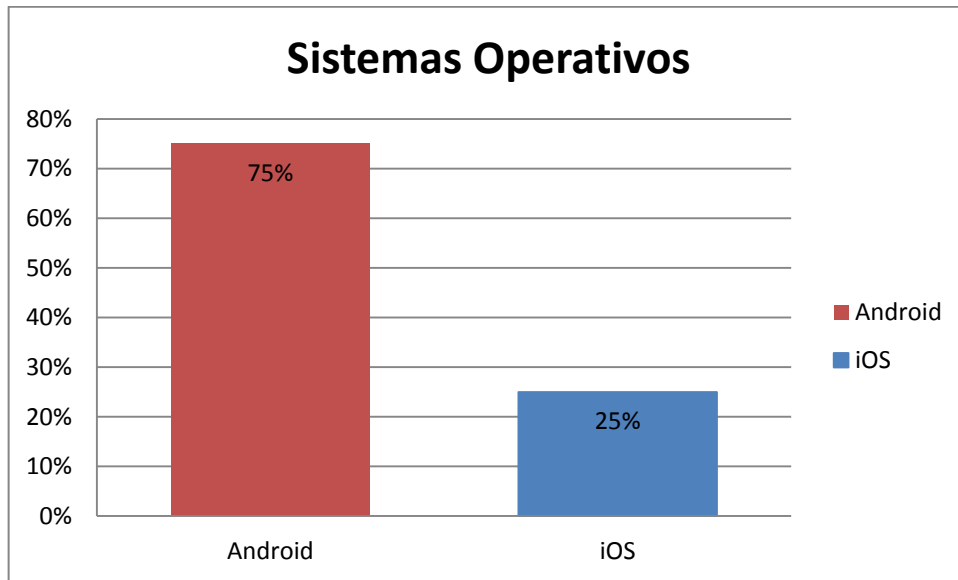


Figura III - 117 Resultado del Parámetro Sistemas Operativos

3.7.3 Interfaz Gráfica de Usuario

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXI Escala de Valoraciones Cualitativa para el Parámetro Interfaz Gráfica de Usuario

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Ninguno	Pocos	Varios	Muchos

Determinación de los Indicadores

TABLA III - XXII Calificación de los Indicadores del Parámetro Interfaz Gráfica de Usuario

INDICADORES	ANDROID	IOS
-------------	---------	-----

Número de formas de definición de la estructura GUI	Muchos	Varios
Numero de formas de fusión de código y diseño	Muchos	Muchos

Fuente: Ver Módulo de Prueba 1

Interpretación de Resultados

$$P_{an} = \sum (V_i)$$

$$P_{an} = \sum (4 + 4)$$

$$P_{an} = 8$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (3 + 4)$$

$$P_{io} = 7$$

$$P_c = \sum (P_{V_i})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (8 / 8) * 100\%$$

$$C_{an} = 100\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (7 / 8) * 100\%$$

$$C_{io} = 87.5\%$$

Interpretación

- En la plataforma Android existen tres formas de definir una interfaz gráfica las cuales son mediante interfaz gráfica del IDE, mediante la definición de código XML, mediante la definición de código Java y para iOS existen dos formas de definir una interfaz gráfica las cuales son mediante la herramienta de diseño de interfaces de usuario Interface Builder y mediante la definición código Objective C. Por lo que para la plataforma Android existen muchas formas de definir una interfaz gráfica y para la plataforma iOS existen varias formas de definir una interfaz gráfica.

- En la plataforma Android existen dos formas de fusionar el código con el diseño las cuales son Mediante la declaración de eventos en el XML de la interfaz gráfica e implementando los eventos de cada elemento mediante código Java; al igual que en Android en iOS existen dos formas de fusionar el código con el diseño las cuales son vinculando de manera visual entre los métodos definidos en la clase controlador y los eventos de cada elemento mediante código Objective C. Por lo que para la plataforma Android y iOS existen muchas formas de fusionar el código con el diseño.

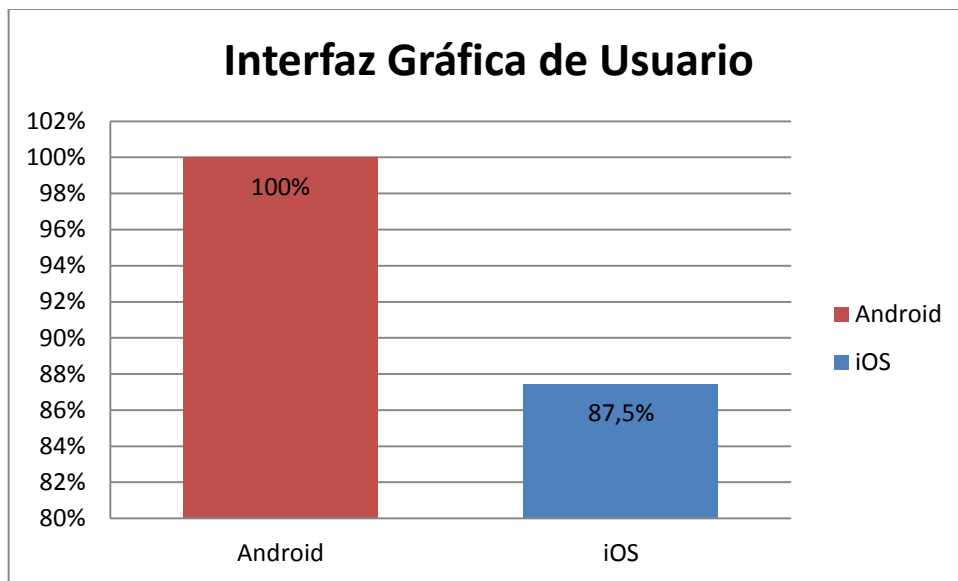


Figura III - 118 Resultado del Parámetro Interfaz Gráfica de Usuario

3.7.4 Persistencia de datos

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXIII Escala de Valoraciones Cualitativa para el Parámetro Persistencia de Datos

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Insuficiente	Parcial	Suficiente	Excelente

Determinación de los Indicadores

TABLA III - XXIV Calificación de los Indicadores del Parámetro Persistencia de Datos

INDICADORES	ANDROID	IOS
Facilidad de inserción de registros en Sqlite	Excelente	Excelente
Facilidad de actualización de registros en Sqlite	Excelente	Excelente
Facilidad de eliminación de registros en Sqlite	Excelente	Excelente
Facilidad de consulta de registros en Sqlite	Excelente	Excelente

Fuente: Ver Módulo de Prueba 1

Interpretación de Resultados

$$P_{an} = \sum (V_i)$$

$$P_{an} = \sum (4 + 4 + 4 + 4)$$

$$P_{an} = 16$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (4 + 4 + 4 + 4)$$

$$P_{io} = 16$$

$$P_c = \sum (P_{V_i})$$

$$P_c = \sum (4 + 4 + 4 + 4)$$

$$P_c = 16$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (16 / 16) * 100\%$$

$$C_{an} = 100\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (16 / 16) * 100\%$$

$$C_{io} = 100\%$$

Interpretación

- En el core de android existen librerías para el manejo nativo de SQLite por lo que la gestión de la base de datos SQLite en la plataforma Android en general es excelente, de igual manera sucede en la plataforma iOS

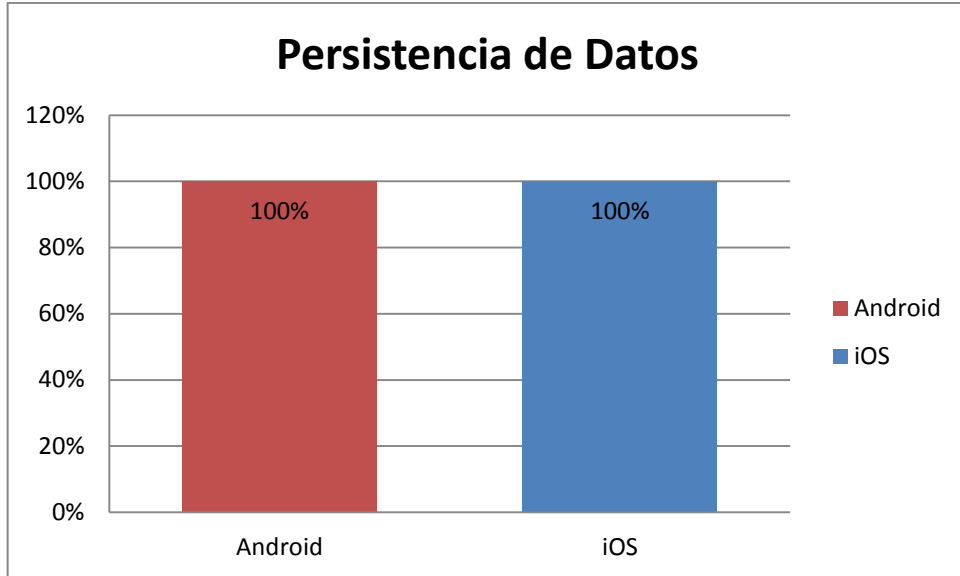


Figura III - 119 Resultado del Parámetro Persistencia de Datos

3.7.5 Acceso a Servicios Web

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXV Escala de Valoraciones Cualitativa para el Parámetro Acceso a Servicios Web

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Insuficiente	Parcial	Suficiente	Excelente

Determinación de los Indicadores

TABLA III - XXVI Calificación de los Indicadores del Parámetro Acceso a Servicios Web

INDICADORES	ANDROID	IOS
Facilidad en el consumo de servicios web utilizando SOAP	Excelente	Suficiente
Facilidad en el consumo de servicios web utilizando REST	Excelente	Suficiente

Fuente: Ver Módulo de Prueba 2

Interpretación de Resultados

$$Pan = \sum (Vi)$$

$$Pan = \sum (4 + 4)$$

$$P_{an} = 8$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (3 + 3)$$

$$P_{io} = 6$$

$$P_c = \sum (P_{Vi})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (8 / 8) * 100\%$$

$$C_{an} = 100\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (6 / 8) * 100\%$$

$$C_{io} = 75\%$$

Interpretación

- En el core de Android existen librerías para consumir servicios web ya sean mediante SOAP o REST, y el código es muy legible, lo cual se realiza de manera excelente.
- En iOS existe librerías para el consumo de servicios web mediante SOAP pero la sintaxis es compleja, en el caso de consumo con REST, iOS no cuenta con librerías propias, sino que debemos añadir librerías externas para poder trabajar con JSON, por lo tanto esto se lo realiza de manera suficiente.

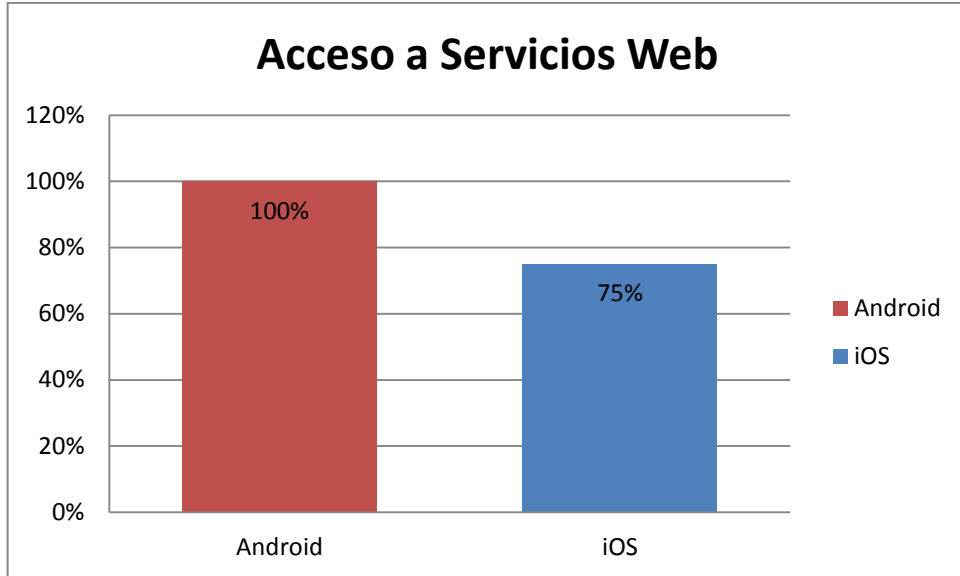


Figura III - 120 Resultado del Parámetro Acceso a Servicio Web

3.7.6 Entorno de Desarrollo Integrado

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXVII Escala de Valoraciones Cualitativa para el Parámetro Entorno de Desarrollo Integrado

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Malo	Regular	Bueno	Muy bueno
Deficiente	Poco eficiente	Eficiente	Muy eficiente

Determinación de los Indicadores

TABLA III - XXVIII Calificación de los Indicadores del Parámetro Entorno de Desarrollo Integrado

INDICADORES	ANDROID	IOS
Capacidad de ayuda inteligente	Muy Bueno	Muy Bueno
Capacidad de Code Snippets	Muy Bueno	Regular
Facilidad de depuración de aplicaciones	Muy Eficiente	Muy Eficiente
Capacidad del diseñador GUI visual	Bueno	Muy Bueno

Fuente: Ver Módulo de Prueba 3

Interpretación de Resultados

$$Pan = \sum (Vi)$$

$$P_{an} = \sum (4 + 4 + 4 + 3)$$

$$P_{an} = 15$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (4 + 2 + 4 + 4)$$

$$P_{io} = 14$$

$$P_c = \sum (P_{Vi})$$

$$P_c = \sum (4 + 4 + 4 + 4)$$

$$P_c = 16$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (15 / 16) * 100\%$$

$$C_{an} = 93.8\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (14 / 16) * 100\%$$

$$C_{io} = 87.5\%$$

Interpretación

- El SDK de Android vinculado con Eclipse nos ofrece una completa ayuda inteligente la misma que nos permite vincularnos a la documentación del código y tener una descripción completa de las clases y métodos, por lo tanto la capacidad de ayuda inteligente es muy buena. De igual manera ocurre con el SDK de iOS vinculado con Xcode
- Los Code Snippets que ofrece Eclipse conjuntamente con el SDK de Android, es muy potente ya que ofrece estructuras de código pequeñas y piezas de código más completas (Por ejemplo una pieza de código para enviar un mensaje) que son útiles en el desarrollo de aplicaciones para android, por lo tanto la capacidad de Code Snippets es muy buena. En Xcode los Code Snippets que ofrece es regular ya que solo son básicos y no existen piezas de código completas.
- En Eclipse y Xcode existen herramientas muy potentes para la depuración de aplicaciones por lo que ambos entornos de desarrollo son muy eficientes.

- EL Diseñador de interfaces de Eclipse, ofrece herramientas básicas para el diseño de interfaces de usuario, con una interfaz no muy bien cuidada. El diseñador de interfaces Interface Builder que esta integrado con Xcode, ofrece herramientas completas para el diseño de interfaces de usuario, con un acabado muy estético y estable.

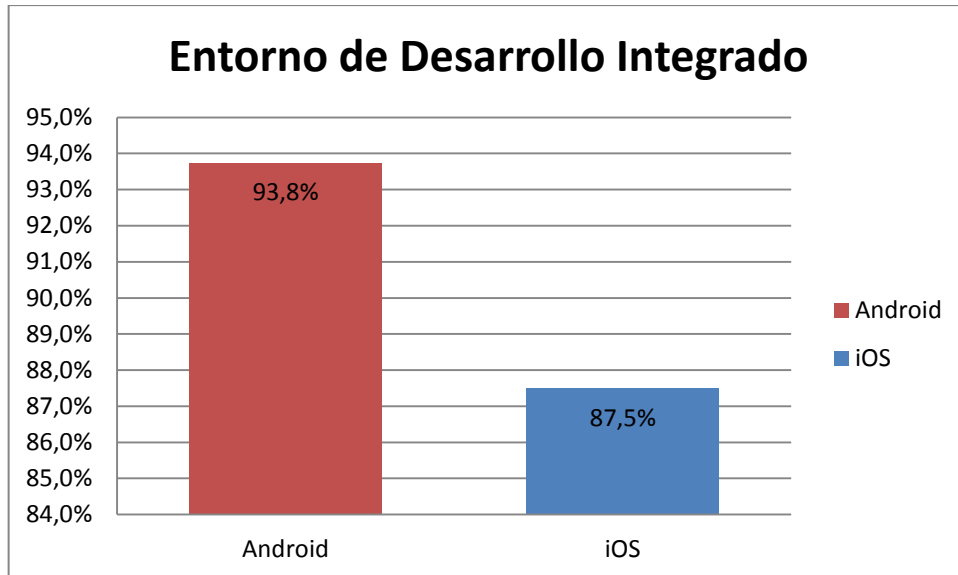


Figura III - 121 Resultado del Parámetro Entorno de Desarrollo Integrado

3.7.7 Compatibilidad entre Versiones

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXIX Escala de Valoraciones Cualitativa para el Parámetro Compatibilidad entre Versiones

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Malo	Regular	Bueno	Muy bueno

Determinación de los Indicadores

TABLA III - XXX Calificación de los Indicadores del Parámetro Compatibilidad entre Versiones

INDICADORES	ANDROID	IOS
Capacidad de manejo de versiones del SDK	Muy Bueno	Muy Bueno
Capacidad de compatibilidad de la aplicación para dispositivos de diferentes	Muy Bueno	Bueno

dimensiones de pantalla		
--------------------------------	--	--

Fuente: Ver Módulo de Prueba 3

Interpretación de Resultados

$$P_{an} = \sum (V_i)$$

$$P_{an} = \sum (4 + 4)$$

$$P_{an} = 8$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (4 + 3)$$

$$P_{io} = 7$$

$$P_c = \sum (P_{V_i})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (8 / 8) * 100\%$$

$$C_{an} = 100\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (7 / 8) * 100\%$$

$$C_{io} = 87.5\%$$

Interpretación

- La forma en que se maneja las versiones del sdk en las dos plataformas Android y iOS es muy buena, ya que permiten a los desarrolladores especificar en que versión del sdk fue desarrollada la aplicación y definir el conjunto de versiones inferiores o superiores del sdk en las que son compatibles.
- Con la versión 4 del sdk de Android da la posibilidad al desarrollador, de crear aplicaciones para dispositivos de diferentes dimensiones de pantalla, es decir la misma aplicación puede ejecutarse en un smartphone y en un tablet con la misma versión del sdk. En la plataforma iOS, una aplicación desarrollada para el iPad no puede ejecutarse en el iPhone por la diferencia en tamaño de pantalla.

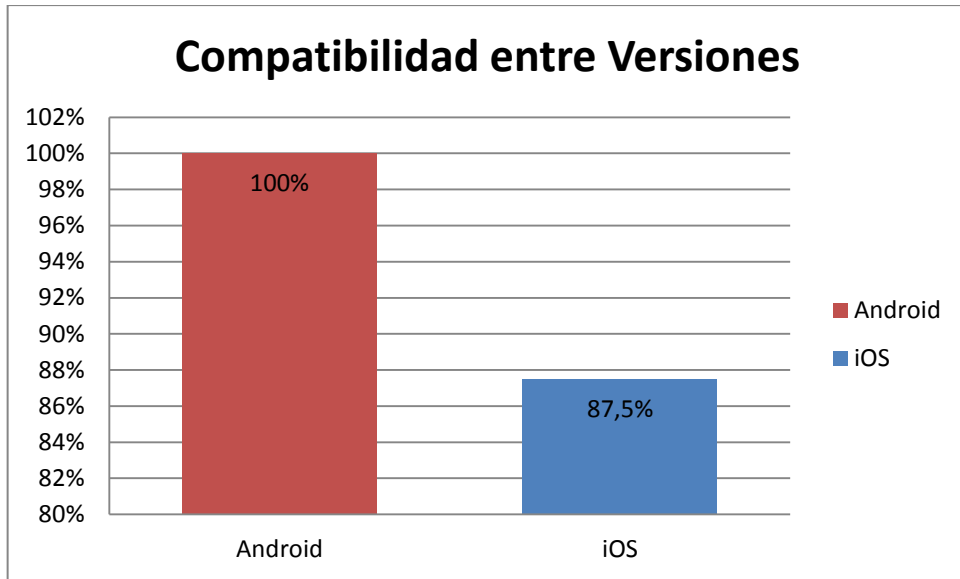


Figura III - 122 Resultado del Parámetro Compatibilidad entre Versiones

3.7.8 Facilidades de Despliegue

Este parámetro se va a evaluar de forma cualitativa, basada en la siguiente tabla:

TABLA III - XXXI Escala de Valoraciones Cualitativa para el Parámetro Facilidades de Despliegue

Valor: 1	Valor: 2	Valor: 3	Valor: 4
Insuficiente	Parcial	Suficiente	Excelente
No satisfactorio	Poco satisfactorio	Satisfactorio	Muy satisfactorio

Determinación de los Indicadores

TABLA III - XXXII Calificación de los Indicadores del Parámetro Facilidades de Despliegue

INDICADORES	ANDROID	IOS
Disponibilidad de emuladores	Excelente	Excelente
Facilidad para desplegar aplicaciones a un dispositivo físico	Muy Satisfactorio	Poco Satisfactorio

Fuente: Ver Módulo de Prueba 3

Interpretación de Resultados

$$Pan = \sum (Vi)$$

$$P_{an} = \sum (4 + 4)$$

$$P_{an} = 8$$

$$P_{io} = \sum (V_j)$$

$$P_{io} = \sum (4 + 2)$$

$$P_{io} = 6$$

$$P_c = \sum (P_{Vi})$$

$$P_c = \sum (4 + 4)$$

$$P_c = 8$$

$$C_{an} = (P_{an} / P_c) * 100\%$$

$$C_{an} = (8 / 8) * 100\%$$

$$C_{an} = 100\%$$

$$C_{io} = (P_{io} / P_c) * 100\%$$

$$C_{io} = (6 / 8) * 100\%$$

$$C_{io} = 75\%$$

Interpretación

- Los emuladores en Android se pueden crear de acuerdo a la versión del sdk y al tamaño de pantalla, en iOS también se puede escoger la versión del sdk y que dispositivo se va a emular (iphone, ipad), por lo tanto la disponibilidad de emuladores es excelente en ambas plataformas para realizar pruebas a las aplicaciones.
- Para desplegar una aplicación en dispositivo físico en la plataforma android es muy simple solo basta con habilitar el modo depuración en el dispositivo, conectarle al computador y ejecutar la aplicación desde Eclipse, por lo que la facilidad de despliegue es muy satisfactoria. En cambio en la plataforma iOS, el proceso no es tan simple, hay que seguir una serie de pasos (ver módulo de prueba 3), que conlleva tiempo y dinero.

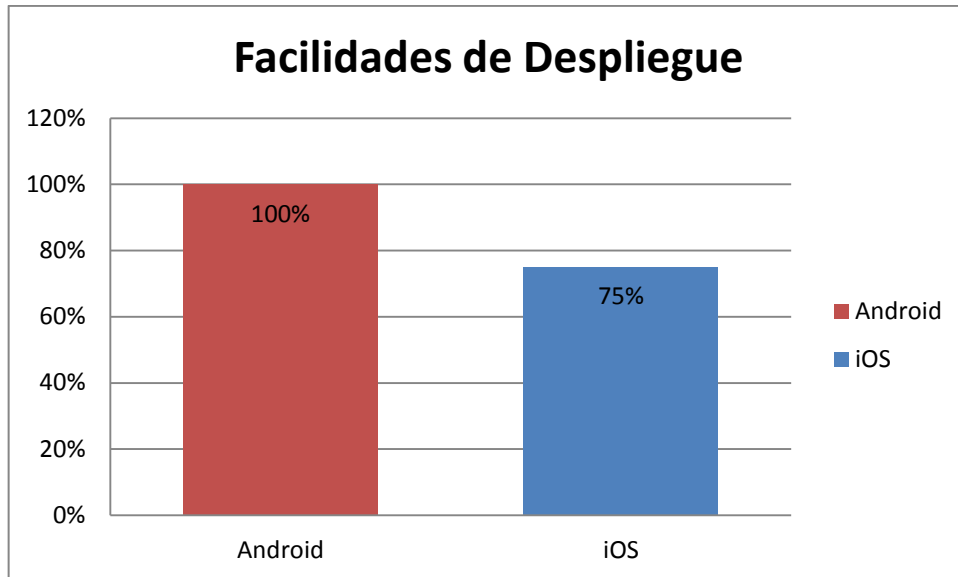


Figura III - 123 Resultado del Parámetro Facilidades de Despliegue

3.7.9 Puntajes Alcanzados

El puntaje final y el porcentaje que ha obtenido cada plataforma se obtiene de la siguiente manera:

Puntaje Total del Análisis

$$PT = \sum (Pc)$$

Puntaje Total de Android

$$PTan = ((\sum(Pan)) / PT) * 100\%$$

Puntaje Total de iOS

$$PTio = ((\sum(Pio) / PT) * 100\%$$

TABLA III - XXXIII Tabla General de Resultados

Parámetro	Indicadores	Tecnologías	
		Android/4	iOS/4
1.- Lenguajes de Programación	1.1.- Cantidad de lenguajes de programación.	2	2
	1.2.- Curva de aprendizaje de los lenguajes de programación.	4	2
2.- Sistemas Operativos	2.1.- Cantidad de sistemas operativos soportados	4	2
	2.2.- Cantidad de sistemas operativos gratuitos	2	0
3.- Interfaz Gráfica de Usuario	3.1.- Numero de formas de definición de la estructura GUI	4	3
	3.2.- Numero de formas de fusión de código y diseño	4	4
4.- Persistencia de datos	4.1.- Facilidad de inserción de registros en Sqlite	4	4
	4.2.- Facilidad de actualización de registros en Sqlite	4	4
	4.3.- Facilidad de eliminación de registros en Sqlite	4	4
	4.4.- Facilidad de consulta de registros en Sqlite	4	4
5.- Acceso a Servicios Web	5.1.- Facilidad en el consumo de servicios web utilizando SOAP	4	3
	5.2.- Facilidad en el consumo de servicios web utilizando REST	4	3
6.- Entorno de Desarrollo Integrado	6.1.- Capacidad de ayuda inteligente	4	4
	6.2.- Capacidad de code snippets	4	2
	6.3.- Facilidad de depuración de aplicaciones	4	4
	6.4.- Capacidad del diseñador GUI visual	3	4
7.- Compatibilidad entre Versiones	7.1.- Capacidad de manejo de versiones del SDK	4	4
	7.2.- Capacidad de compatibilidad de la aplicación para dispositivos de diferentes dimensiones de pantalla	4	3
8.- Facilidades de Despliegue	8.1.- Disponibilidad de emuladores	4	4
	8.2.- Facilidad para desplegar aplicaciones a un dispositivo físico	4	2
TOTAL		75/80	62/80

3.8 Diagrama General De Resultados

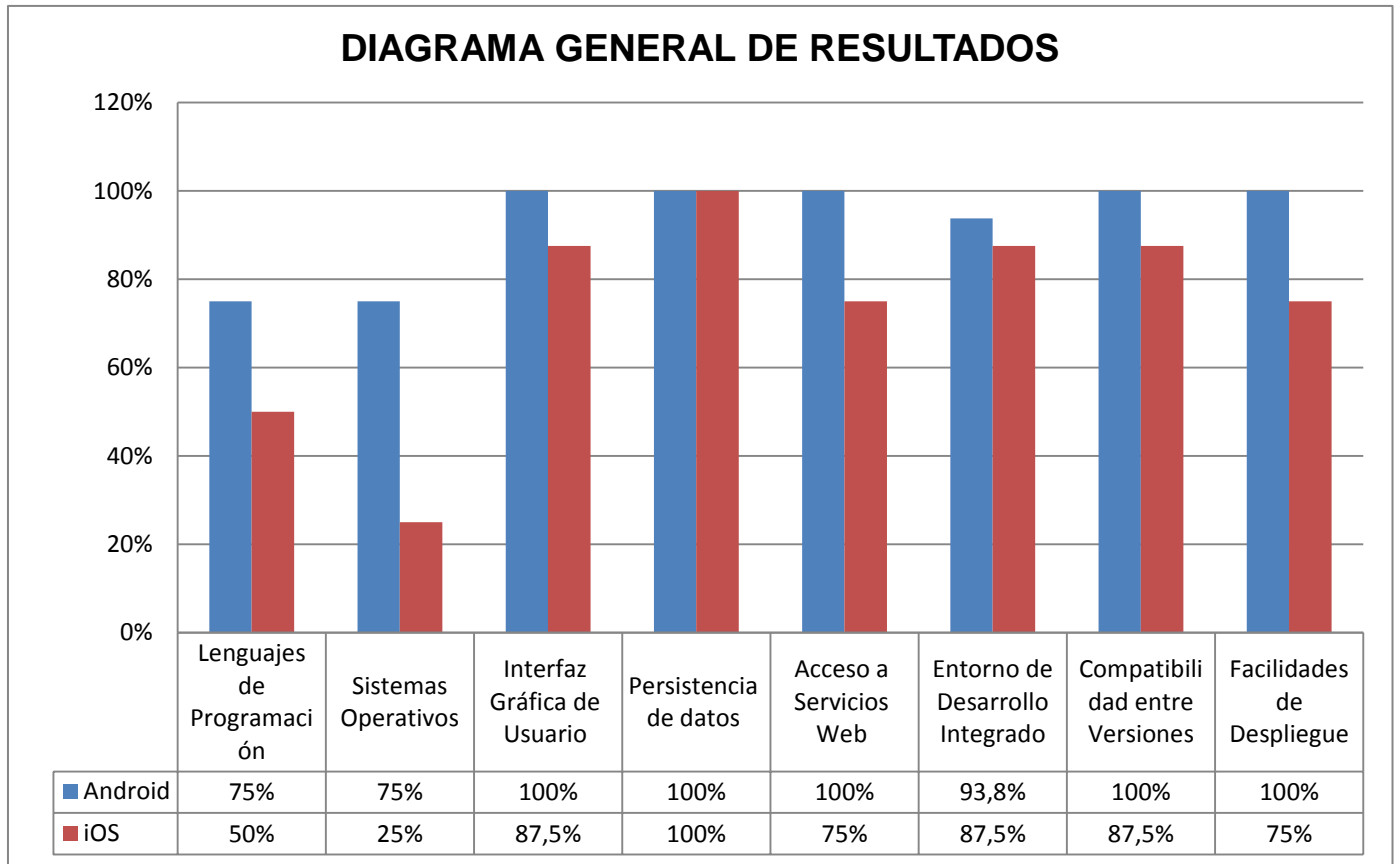


Figura III - 124 Gráfico General de Resultados

Puntaje Total del Análisis

$$PT = \sum (Pc)$$

$$PT = 80$$

Puntaje Total de Android

$$PTan = ((\sum(Pan)) / PT) * 100\%$$

$$PTan = (75 / 80) * 100\%$$

$$PTan = 93.75 \%, \text{ Equivalente a Muy bueno}$$

Puntaje Total de iOS

$$PTio = ((\sum(Pio)) / PT) * 100\%$$

$$PTio = (62/80) * 100\%$$

$$PTio = 77.5 \%, \text{ Equivalente a Bueno}$$

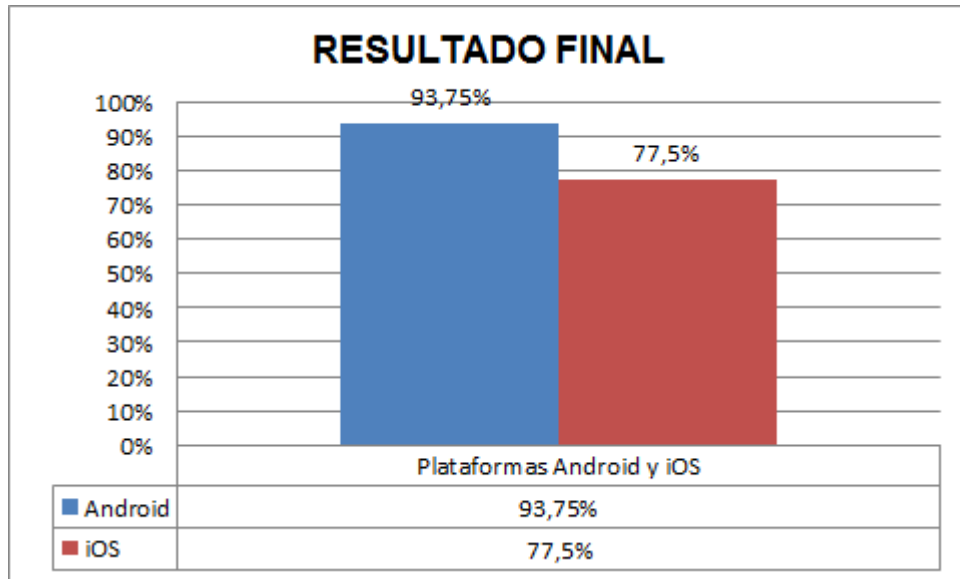


Figura III - 125 Gráfico General de Resultados

3.8.1 Interpretación

De acuerdo al análisis y a los puntajes obtenidos con cada una de las plataformas de desarrollo de aplicaciones para los sistemas operativos android y iOS se ha obtenido como resultado que la plataforma Android ha obtenido el puntaje más alto con un porcentaje del 93.75% que es equivalente a Muy Bueno, Mientras que la plataforma iOS ha obtenido un valor de 77,5% equivalente a Bueno, por lo tanto se deduce que el puntaje obtenido en la plataforma Android es mayor que en la plataforma iOS.

3.8.2 Resultado del Análisis

Luego del análisis correspondiente realizado a las plataformas de desarrollo de aplicaciones para los sistemas operativos android y iOS, podemos determinar lo siguiente:

El lenguaje de programación Java usado por defecto para el desarrollo de aplicaciones en Android es muy potente y presenta a los desarrolladores una curva de aprendizaje muy buena por su sintaxis comprensible, mientras que el lenguaje de programación objective C tiene una sintaxis compleja

La plataforma Android puede ser instalada en sistema operativos propietarios y libres, esto hace que existan más desarrolladores en Android, mientras que la plataforma iOS solo se ejecuta en sistema operativo Mac OS X.

- Con relación a la interfaz gráfica de usuario, ambas plataformas de desarrollo de aplicaciones Android y iOS presentan elementos propios para el uso en interfaces táctiles.
- La gestión de base de datos SQLite en ambas plataformas es excelente debido a que existen librerías nativas para su manejo y esta base de datos utiliza sentencias sql, las mismas que son conocidas por los programadores.
- En la base de datos SQLite que viene en Android, las operaciones básicas en una base de datos como la inserción, actualización, eliminación y consulta de datos es bastante simple. En iOS estas operaciones son también bastante simples la diferencia está en que las realiza con un número mayor de líneas de código. Esta diferencia no es tan grande ya que en ocasiones se diferencia en una línea de código.
- El consumo de servicios web en la plataforma android, es relativamente fácil, por las librerías nativas que trae para este proceso, mientras que en la plataforma iOS hay que utilizar librerías externas y su utilización depende de cómo fueron implementadas.

- Los IDEs Eclipse y Xcode que son utilizados para el desarrollo de aplicaciones para android y iOS respectivamente, tienen herramientas completas para codificación y diseño de las aplicaciones muy eficientes.
- En ambas plataformas existen emuladores muy eficientes para realizar pruebas a nuestras aplicaciones, para el despliegue de aplicaciones ya en un dispositivo físico en la plataforma Android es muy fácil por ser software libre, mientras que para iOS el despliegue ocupa un poco más de tiempo y hay que pagar por instalar las aplicaciones en el dispositivo.

3.9 Comprobación de Hipótesis

La hipótesis planteada es:

Hi: Una plataforma de desarrollo para dispositivos móviles inteligentes sobre el sistema operativo android permitirá una mayor productividad que en el sistema operativo iOS en la implementación de aplicaciones.

Variable Independiente: Plataforma de desarrollo para dispositivos móviles inteligentes sobre el sistema operativo android.

Variable Dependiente: Productividad en la implementación de aplicaciones

La productividad se mide en base a los parámetros e indicadores (Tabla III-XXXIII) y haciendo referencia a la Figura III-125 “Grafico General de Resultados” y por observación directa se concluye que en la hipótesis planteada: Android posee el valor más alto 75 puntos sobre 80, valor máximo posible, superando en un 16,25% a iOS, Android tiene el nivel de productividad del 93.75%/100. Por lo tanto se concluye que la hipótesis H1 es verdadera.

3.9.1 Conclusión

Se concluye que la plataforma de desarrollo para dispositivos móviles inteligentes sobre el sistema operativo android es la más adecuada para el desarrollo de nuestra aplicación, puesto que ha tenido un mejor desenvolvimiento en los diferentes parámetros descritos en este trabajo de Investigación, principalmente en los parámetros como lenguajes de programación, acceso a servicios web y facilidades de despliegue, en los cuales Android tiene mejores prestaciones con relación a iOS.

CAPITULO IV

DESARROLLO DE LA APLICACIÓN PARA ATENCIÓN DE CLIENTES EN LA RECOLECCIÓN DE PEDIDOS EN BONNY RESTAURANT

4.1. PARTE TEÓRICA

4.1.1. Introducción

Los dispositivos móviles en Ecuador aún no tienen una aplicación en el ámbito de atención al cliente, en donde las colas de espera y la experiencia de las personas puedan ser mejoradas, con un beneficio directo no sólo para el consumidor, sino también para los dueños del establecimiento prestador del servicio.

En los restaurantes del país los clientes usualmente tienen que abrir la carta tradicional para seleccionar su orden y respetar con paciencia características “estándar” de atención al cliente, como las siguientes:

- Realizar su pedido de una manera tradicional hacia un mesero que se acerque a solicitar la orden.
- Existencia de una cola de personas para realizar su pedido.
- Existencia de una cola de clientes para realizar su pago.
- Inexistencia de servicio para la reservación de mesas.

- Inexistencia de información de la disponibilidad de los platos.
- Inexistencia de información detallada acerca de las comidas disponibles (foto, descripción, especies, etc.).

Características adicionales para los dueños del local

- No se lleva estadísticas de los platos más vendidos.

Características indirectas

- Falta de motivación para que el cliente acceda al servicio.

En este contexto no existe una buena estrategia para motivar al cliente a que acceda al servicio de restaurante.

En este ámbito, con la ayuda de los dispositivos móviles de bajo costo y el uso de tecnología de comunicación inalámbrica, se puede desarrollar una aplicación software que ayude a este proceso de atención al cliente.

SIGRES es un sistema que brinda las acciones necesarias para la administración de un restaurante, y atención del cliente en la recolección de pedidos.

4.1.2. Desarrollo

El sistema informático está desarrollado en una metodología ágil: Programación Extrema XP, que se basa en una serie de reglas y principios de la Ingeniería de Software tradicional, pero con la excepción de que XP da prioridad a las tareas que es lo que precisamente da resultados directos y reduce los procesos largos de recopilación de gran información para cumplir los resultados planificados.

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos,

la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto.

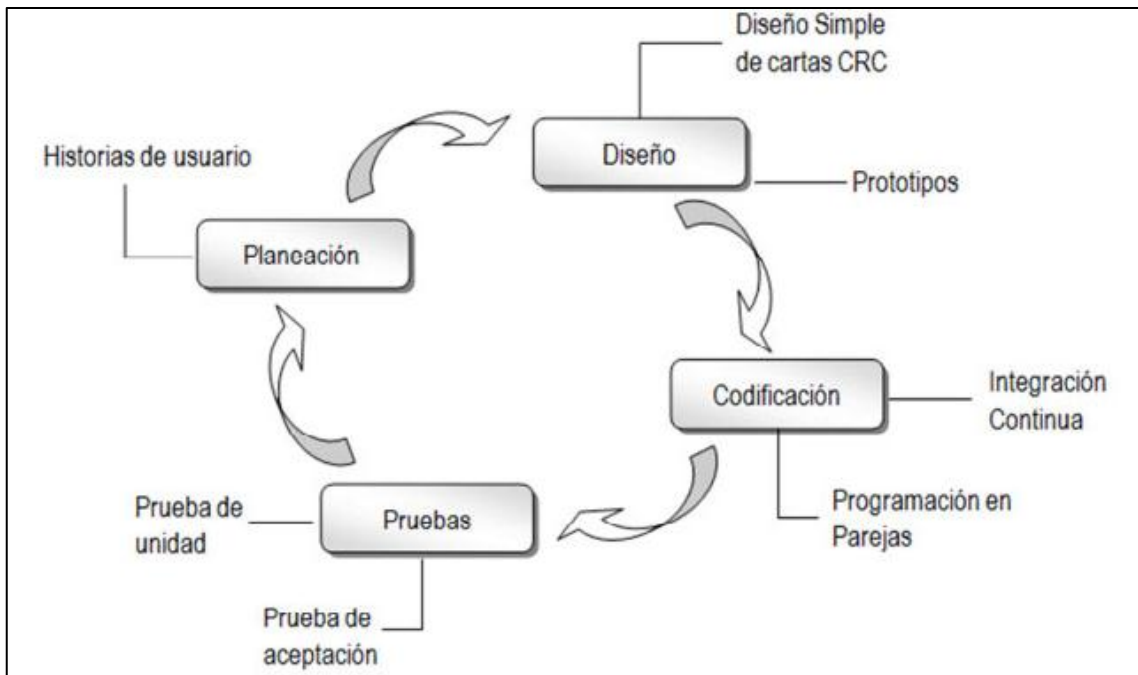


Figura IV - 1 Usuario Metodología XP

Se va a abarcar todo el ciclo de vida, empezando por los requisitos y acabando en el sistema funcionando, proporcionando así una visión completa y coherente de la producción de sistemas software. El enfoque que toma es el de un ciclo de vida iterativo incremental, el cual permite una gran flexibilidad a la hora de adaptarlo a un proyecto y a un equipo de desarrollo específicos. El ciclo de vida está dirigido por casos de uso, es decir, por la funcionalidad que ofrece el sistema a los futuros usuarios del mismo. Así no se pierde de vista la motivación principal que debería estar en cualquier proceso de construcción de software: el resolver una necesidad del usuario/cliente.

I. Planificación

- Historias de usuario.
- Plan de Publicaciones.
- Iteraciones:
- Velocidad del proyecto.
- Programación en pareja.
- Reuniones diarias.

II. Diseño

- Diseños simples.
- Glosarios de términos.
- Riesgos.
- Funcionalidad extra.
- Tarjetas C.R.C.

III. Codificación

IV. Pruebas

- El uso de los test en X.P es el siguiente.
- Test de aceptación.

I. Planificación

- **Historias de usuario.**

El primer paso en un proyecto que utilice la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles

- **Plan de Publicaciones.**

Es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la

prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa.

- **Iteraciones**

Se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el Plan de Publicaciones que serán implementadas.

- **Velocidad del proyecto.**

La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración.

- **Programación en pareja.**

El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.

- **Reuniones diarias.**

Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

II. Diseño

- **Diseños simples.**

Se debe hacer de una manera simple para conseguir un diseño fácilmente entendible para que la implementación a la larga ocupe menos tiempo y esfuerzo en el desarrollo.

- **Glosarios de términos.**

Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

- **Riesgos.**

Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

- **Funcionalidad extra.**

No se debería añadir funcionalidad extra al programar, aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

- **Tarjetas C.R.C.**

El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a

objetos olvidándose de los malos hábitos de la programación procedural clásica.

III. Codificación

La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

X.P también propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo.

IV. Pruebas

Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.

Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

4.2. PARTE PRÁCTICA

4.2.1. Herramientas utilizadas

Se optó por seleccionar herramientas libres para el desarrollo de la aplicación. Por un lado se empleó el SDK de Android como herramienta de desarrollo para la aplicación cliente mientras que para la aplicación servidor se empleó el framework symfony y como motor de base de datos se decidió por MySQL. A continuación se detalla cada una de éstas.

- **Android SDK**

El SDK (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono, documentación, ejemplos de código y tutoriales.

- **Eclipse**

Es una herramienta para el programador diseñada principalmente para el desarrollo de aplicaciones Java, facilitando al máximo la gestión de proyectos colaborativos mediante el control de versiones', es posible también con subversión, exportar e importar proyectos.

- **Symfony**

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

- **Netbeans**

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

- **Mysql**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

4.2.2. Introducción

La implementación de un sistema informático para la recolección de pedidos que realizan los clientes en el restaurante Bonny Restaurant, estará formado por una aplicación web que permitirá realizar las actividades básicas en la gestión del restaurante, en donde estarán publicados servicios web que serán consumidos desde la aplicación cliente desarrollada en android.

4.2.3. Definición del Ámbito

Bonny Restarant, es una empresa dedicada al servicio en el sector gastronomía, ofrece platos y bebidas, para ser consumidas en el mismo local o para llevar. Es posible escoger uno mismo una gran variedad de platos cocinados y dispuestos para el autoservicio. A veces se paga una cantidad fija y otras veces por cantidad consumida.

El chalet con una capacidad máxima para 150 personas, y La viejoteca con capacidad para 60 personas le permiten organizar:

- Eventos Sociales: Matrimonios, 15 años, Bautizos, Fiestas de Grado, etc.
- Seminarios y reuniones empresariales, mesas de negocios.
- Almuerzos y Cenas Institucionales.

4.2.4. Equipo de Trabajo

Director:

Ing. Danilo Pastor.

Miembros:

Ing. Danilo Pastor

Ing. Gloria Arcos

Srta. Karla Carvajal

Sr. Christian Sasig

4.2.5. Servicios

- Ofrecer una gran variedad de platos a sus clientes para diferentes ocasiones.
- Ofrecer bebidas especiales de acuerdo a cada ocasión.
- Dar acceso libre y gratuito a internet mediante su zona wifi que cubre todo el restaurante.
- Da a disposición un Salón de Eventos multiuso, para grandes y medianas empresas que son usados para ofrecer un ShowRoom a sus clientes estratégicos de la Región
- Servicio de Recepciones y Catering y Cenas Románticas

4.2.6. Funciones

- Dar a conocer los diferentes platos que ofrece el restaurante.
- Dar a los clientes la opción para escoger la mesa de su preferencia.
- Hacer que la experiencia del cliente sea la mejor al usar su servicio.
- Realizar recepciones y Catering y Cenas Románticas.

4.2.7. Descripción del Sistema Actual

La forma actual en que se ofrece los variados platos a los clientes es que el mesero les entrega la carta tradicional (esta carta no tiene una descripción detallada del producto ni los ingredientes), para que los clientes escojan sus comidas, el mesero recibe la orden y la entrega a la cocina, luego le sirven al cliente su orden. Esto se lo realiza la mayoría de restaurantes por lo que la experiencia al hacer uso de este servicio es igual a los demás restaurantes, y además no tienen un reporte acerca de sus platos más vendidos en una fecha determinada.

4.2.8. Identificación del Problema

El problema de forma general se presenta en que el cliente no conoce exactamente que se va a servir y no existe información actualizada de los platos.

4.2.9. Fases

4.2.9.1. Planeación

Para cada uno de los elementos se enuncia lo que la teoría sobre XP recomienda contrastándola con la experiencia real en la realización del proyecto. Entre los elementos a discutir para esta fase se encuentran las historias de usuario, el plan de entregas, lo relacionado con las iteraciones como las modificaciones que se aplicaron a XP para hacerla más adecuada para el proyecto.

4.2.9.1.1 Historias de Usuario

Si bien el cliente no fue quien escribió personalmente las historias de usuario (Ver Anexo 1), fue él quien diseñó su contenido y dirigió la redacción de las mismas, debido a que no tenía los conocimientos necesarios en formato para elaborarlas. A pesar de lo anterior, el propósito de las mismas no se vio alterado de alguna forma, manteniendo no solamente la terminología del cliente

al punto en que este fuera autosuficiente en la comprensión de su contenido, sino también su oficio como punto de partida en la planificación del proyecto.

Desde el punto de vista del nivel de detalle, se siguió la directiva en el sentido de no profundizar ni en descripciones ni en procesos, manteniéndolas de estas formas breves y claras. Sin embargo se logró abstraer la información suficiente de ellas para realizar su implementación sin requerir demasiadas aclaraciones por parte del cliente, siendo factor fundamental para no ocasionar retrasos motivados por falta de claridad en los requerimientos.

Por otro lado es muy importante resaltar el papel fundamental que jugaron las historias de usuario en la estimación de los tiempos requeridos para el desarrollo del proyecto. Una vez recolectadas todas las historias de usuario, se hizo una reunión del equipo de trabajo donde se plantearon los tiempos necesarios para su implementación, los cuales resultaron en estimaciones inusualmente aproximadas de los tiempos de desarrollo en comparación con los realmente requeridos. Esto es importante resaltarlo debido al poco nivel de detalle que las historias de usuario tenían, significando la poca información sobre las implicaciones técnicas de su implementación.

4.2.9.1.2 Plan de Publicaciones

Debido a que las iteraciones tenían una duración de 15 días, fue al término de este plazo que se realizaron entregas, las cuales siempre fueron funcionales, lo que quiere decir que al momento de la entrega estaban en condiciones de ser puestas en funcionamiento en las instalaciones del cliente. Esto representó un éxito en el desarrollo del proyecto ya que mantenía el interés del cliente en continuarlo debido a que estaba viendo resultados en el corto plazo.

4.2.9.1.3 Velocidad del proyecto

El número de historias de usuario realizadas por iteración no fue una buena medida de la velocidad del proyecto debido que no todas tenían el mismo nivel de dificultad y por tanto el mismo requerimiento de horas de desarrollo. Por

esto se encontró que mientras en la segunda iteración se trabajaron menos horas semanales en comparación con las demás, también fue donde más historias de usuario se evacuaron, lo que supondría un nivel de rendimiento muy superior, lo que no es cierto. El motivo de este resultado fue que el nivel de dificultad y por lo tanto, el número de horas requeridas para las historias de usuario de la segunda iteración fue el más bajo de todo el proyecto.

TABLA IV - I Velocidad del Proyecto

	Iteración 1	Iteración 2	Iteración 3
Horas	46	41	42
Semanas	4	4	2
Horas semanales	23	25	25
Historias de usuario (Velocidad del proyecto)	5	6	6

Si bien esta medida de velocidad del proyecto fue tomada en cuenta para el análisis de tiempos, resultó de mayor utilidad estimar el número de horas que tomaría implementar cada historia de usuario y planificar las entregas acorde con esta medida. De esta forma, al tener la disponibilidad de cada desarrollador en horas por semana, se pudo estimar con mucha precisión cuántas historias de usuario podían ser asignadas en iteración. Esta medida de la velocidad del proyecto resultó tan acertada que permitió realizar un plan de entregas preciso, y lo más importante, cumplirlo.

4.2.9.1.4 Iteración

El desarrollo de la aplicación se detalla mediante iteraciones y los datos obtenidos en cada una de ellas, mediante la cual nos permite la depuración de los requerimientos y las dificultades que se presentan en el desarrollo.

Iteración 1

Recopilación de Información necesaria para desarrollar el proyecto de tesis.

Iteración 2

Análisis de la base de datos Mysql, el framework Symfony 2 y el SDK de Android para el desarrollo del proyecto de tesis.

Iteración 3

Generación del primer prototipo funcional, análisis de los problemas presentados y documentación del proceso.

Análisis de resultados

Iteración 1

Como resultado de la primera iteración se obtuvo una propuesta para el desarrollo del proyecto de tesis.

Iteración 2

Como resultado de esta iteración se obtuvo una conexión exitosa con la base de datos Mysql con el proyecto creado con Symfony.

Iteración 3

Como resultado se obtiene el prototipo funcional del sistema gestión de restaurantes versión 1.0.

4.2.9.1.5 Programación en parejas

El objetivo que ambos miembros del grupo conocieran cada aspecto de la implementación fue fácil de lograr para el proyecto, debido a que por un lado solo se tenían dos desarrolladores y por el otro que el tamaño del proyecto así lo permitió.

4.2.9.2. Diseño

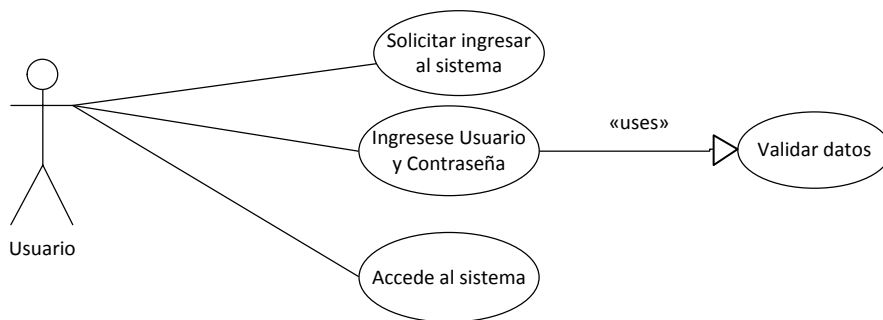
4.2.9.2.1 Diseños Simples

Una de las partes más importantes de la filosofía XP es la simplicidad en todos los aspectos. Se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo, por lo cual esto es lo que se busca. La idea es que se haga el diseño más sencillo que cumpla con los requerimientos de las historias de usuario.

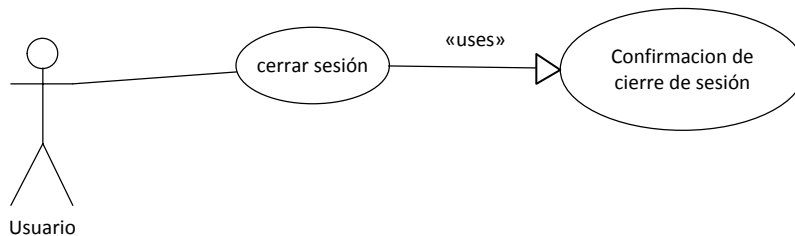
Sobre los diagramas, se es muy claro que se pueden usar siempre que no tome mucho tiempo en realizarlos, que sean de verdadera utilidad y que se esté dispuesto a “tirarlos a la basura”. En XP se prefiere tener una descripción del sistema o parte de él, en lugar de una serie de complejos diagramas que probablemente tomen más tiempo y sean menos instructivos

Refinamiento de los casos de uso.

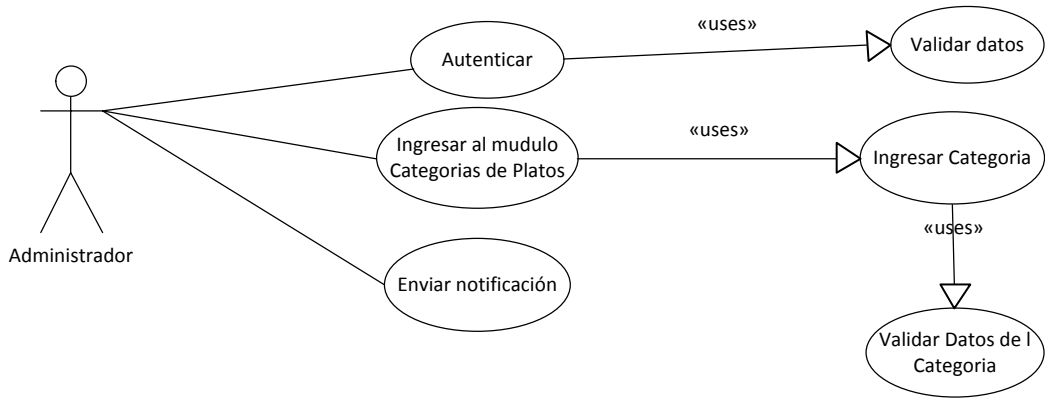
Caso de uso Autenticar



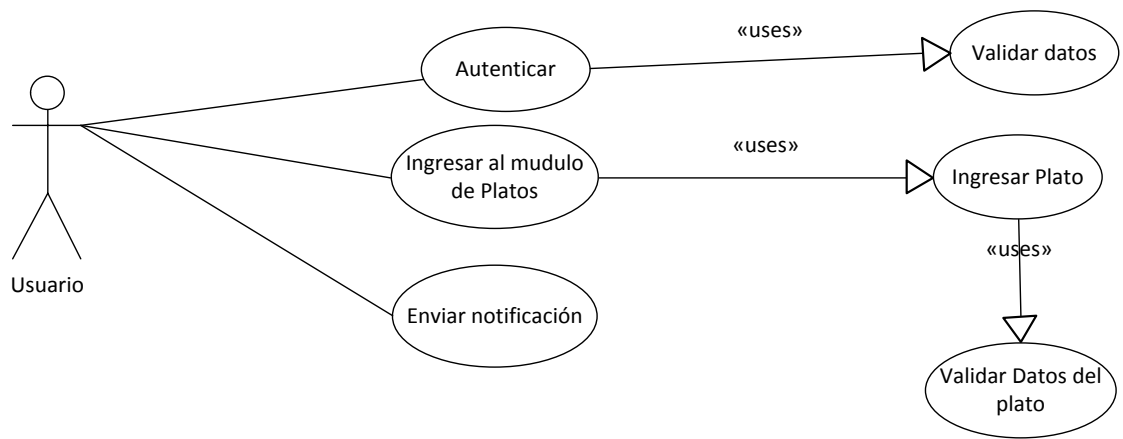
Caso de Uso Cerrar Sesión



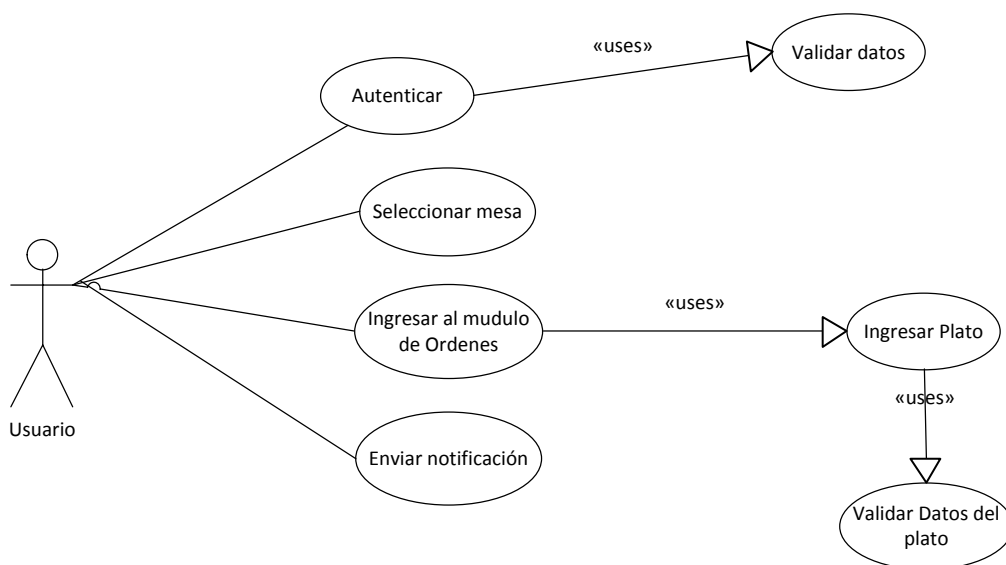
Caso de Uso Agregar Categorías de Platos



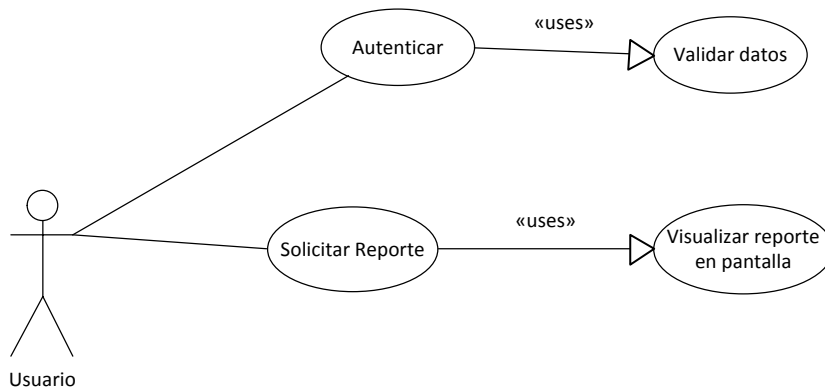
Caso de Uso Agregar Platos



Caso de Uso Agregar Orden



Caso de Uso Generar Reporte



Definición e interacción de formularios para la interfaz de usuario

Aplicación Servidor

Página Principal

Administración

Bienvenido(a): Karlita Carvajal - Super Administrador Salir

Gestión de Menus : Menus

Previo Nuevo Ayuda

Filtro: Buscar Limpiar Grupo de menu: Seleccione un grupo de menu

Nombre del menu: Todos los menus Grupo de ingredientes: Seleccione un grupo de ingre

No.	Nombre	Descripción	Precio	Stock	Stock mínimo	Estado	Grupo de ingredientes	Acciones
1	Camarones al Ajillo	Este plato es autoctono de la costa ecuatoriana, su ingrediente principal son los mariscos y pescados del mar.	7.00	20	5	✓		
2	Ceviche Peruano	Este plato es autoctono de la costa ecuatoriana, su ingrediente principal son los mariscos y pescados del mar.	8.00	15	6	✓		
3	Conchas Asadas	Este plato es autoctono de la costa ecuatoriana, su ingrediente principal son los mariscos y pescados del mar.	10.00	50	10	✓		
4	Trucha Frita	Este plato es autoctono de la costa ecuatoriana, su ingrediente principal son los mariscos y pescados del mar.	5.00	10	5	✓		

Mostrar # 20 Inicial Previo 1 Siguinte Fin Página 1/1 Total 4

Figura IV - 2 Listado de Platos

The screenshot shows the 'Gestión de Ordenes : Ordenes' interface. It includes a sidebar with navigation options like 'Sitio', 'Gestión', 'Clientes', 'Ordenes', 'Usuarios', and 'Reportes'. The main area features search filters for 'Cédula', 'Fecha de inicio', 'Fecha de fin', 'Estado de Pago', 'Mesa', and 'Estado de la Orden'. Below the filters is a table with the following data:

No.	No. de orden	Nombre del Cliente	Mesa	Fecha de Solicitud	Fecha de Ejecución	Total	Estado de Pago	Estado de la Orden	Acciones
1	1	Karla Carvajal (0604836837)	Mesa Central Interior	25/10/2011	14/10/2011	44.80	No	Pendiente	[Icons]
2	2	Karla Carvajal (0604836837)	Mesa Central Interior	27/01/2012	27/01/2012	2.96	No	Pendiente	[Icons]
3	3	Karla Carvajal (0604836837)	Mesa Central Interior	27/01/2012	27/01/2012	12.81	No	Pendiente	[Icons]
4	4	Karla Carvajal (0604836837)	Mesa Central Interior	27/01/2012	27/01/2012	5.91	No	Pendiente	[Icons]
5	5	Karla Carvajal (0604836837)	Mesa Central Interior	27/01/2012	27/01/2012	11.83	No	Pendiente	[Icons]

Figura IV - 3 Listado de ordenes

The screenshot shows the 'Gestión de Usuarios : Usuarios' interface. It includes a sidebar with navigation options like 'Sitio', 'Gestión', 'Clientes', 'Ordenes', 'Usuarios', and 'Reportes'. The main area features a search filter for 'Nombre del usuario'. Below the filter is a table with the following data:

No.	Nombre	Nombre de acceso	Email	Estado	Acciones
1	Karla Carvajal	superadmin	cs@yahoo.es	✓	[Icons]
2	user	user	cs2@yahoo.es	✗	[Icons]

At the bottom of the table, there is a pagination control: 'Mostrar # 20', navigation buttons for 'Inicio', 'Previo', '1', 'Siguiete', 'Fin', and 'Página 1/1 Total 2'.

Figura IV - 4 Listado de Usuarios

Aplicación Cliente

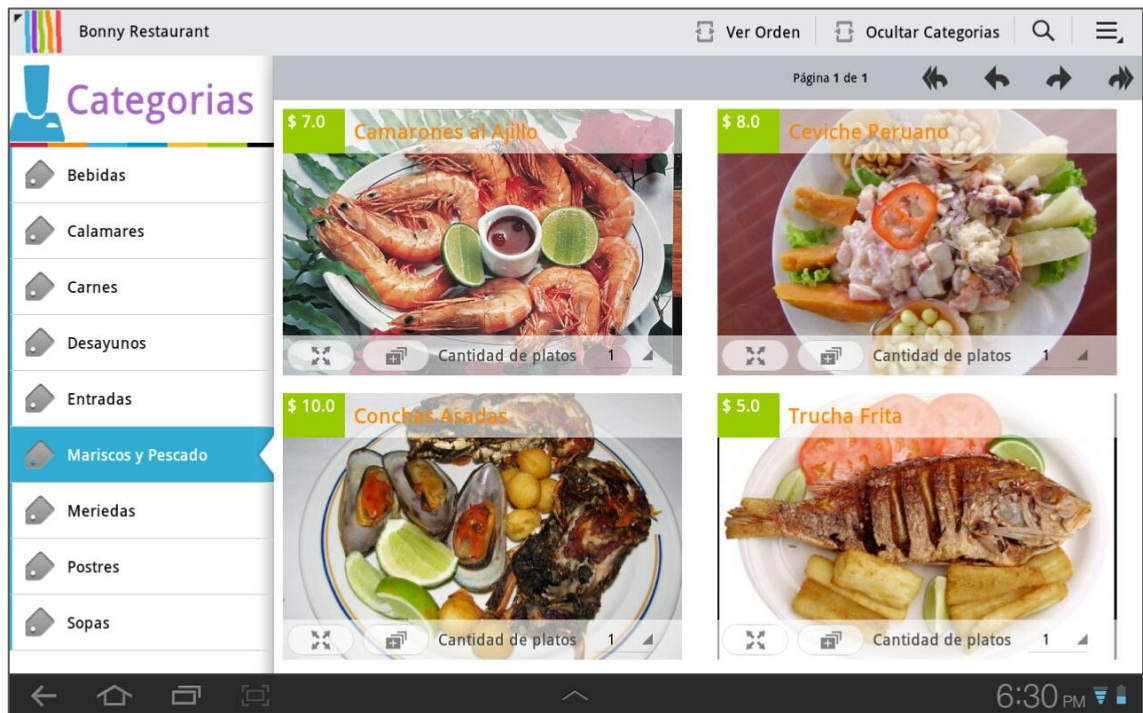


Figura IV - 5 Navegación por las diferentes categorías



Figura IV - 6 Detalle de un plato

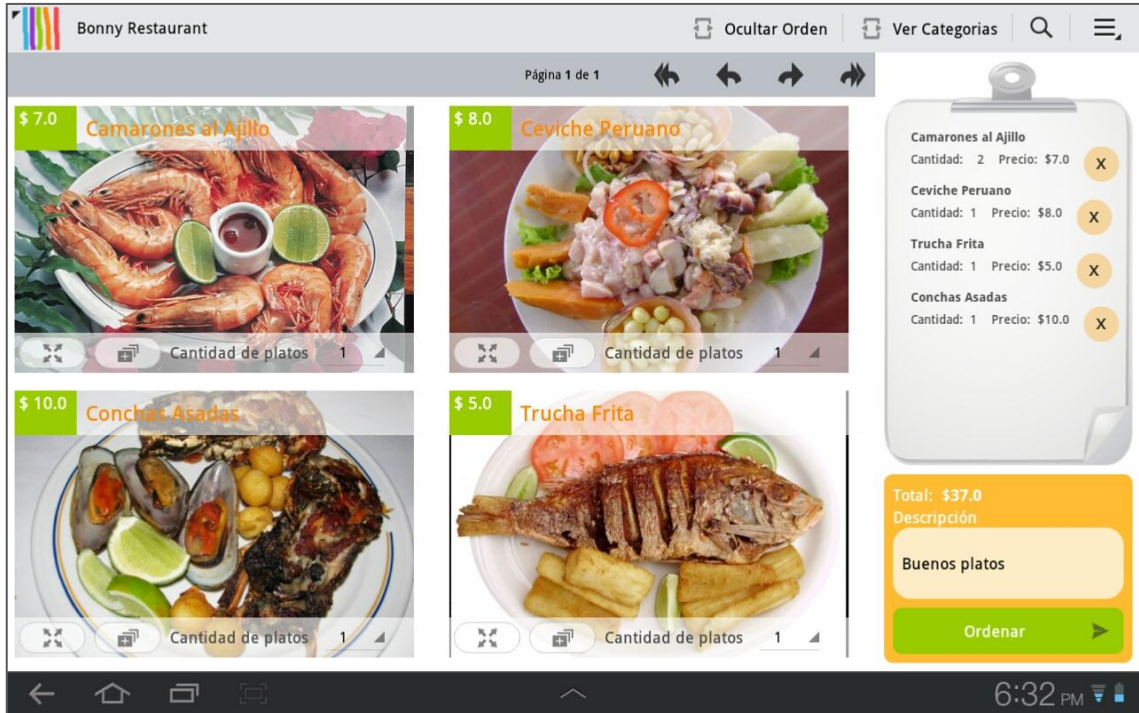


Figura IV - 7 Detalle de la orden

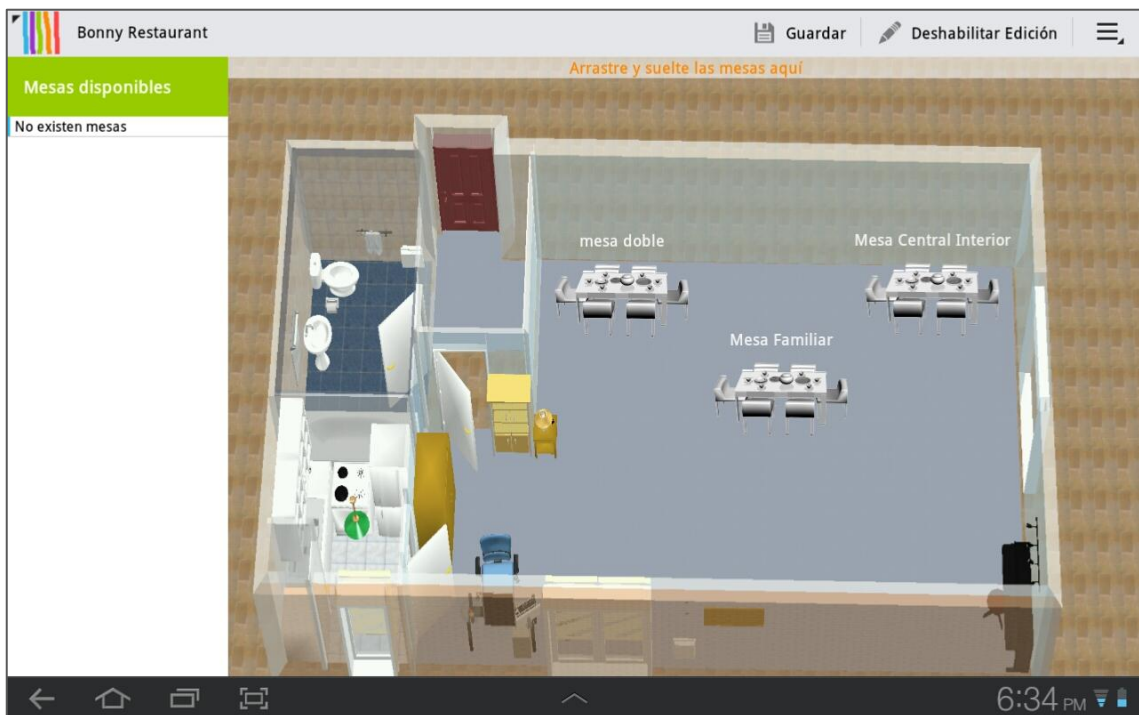


Figura IV - 8 Ubicación de mesas

Diagrama de Componentes

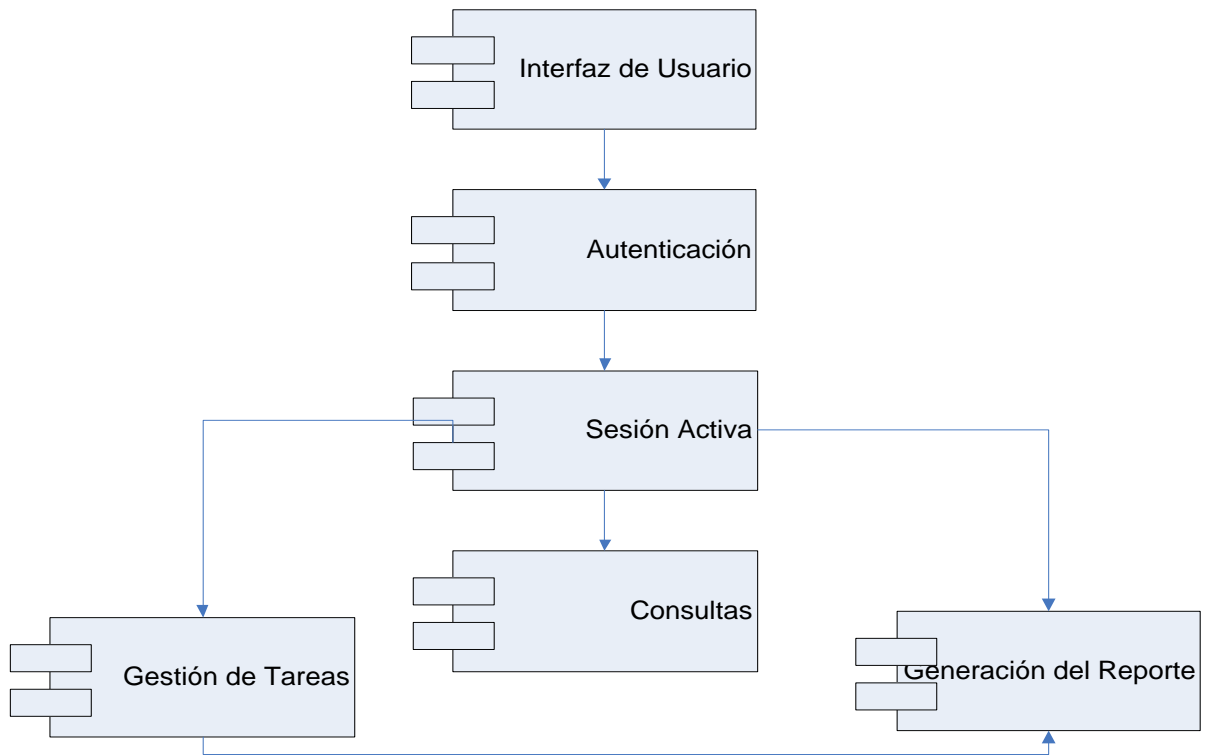


Figura IV - 9 Diagrama de componentes

Arquitectura Física de la Aplicación

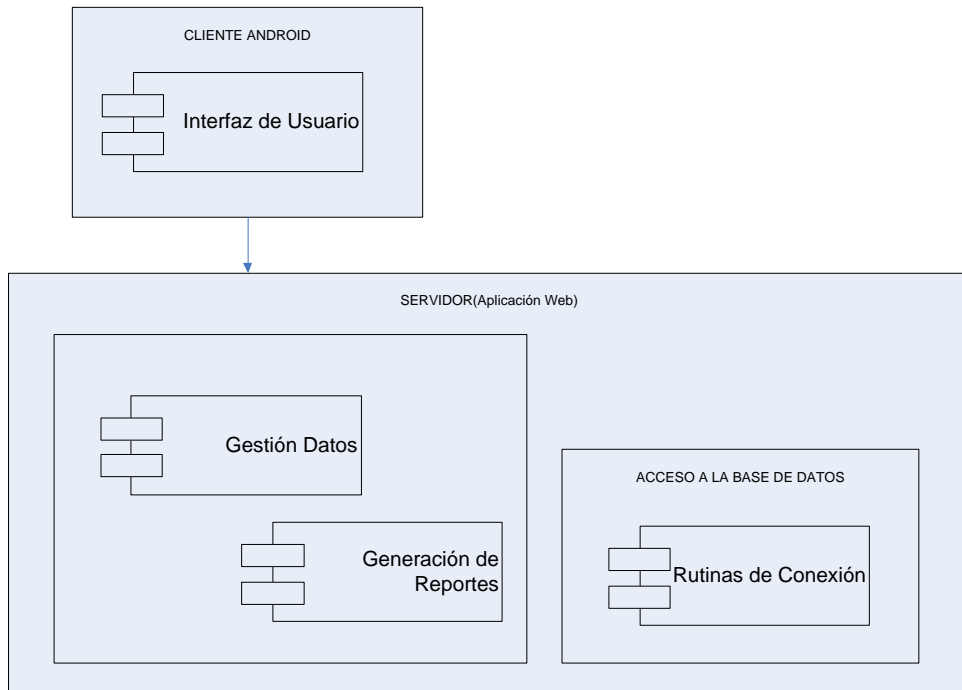


Figura IV - 10 Arquitectura física de la aplicación

Diseño de la Base de Datos

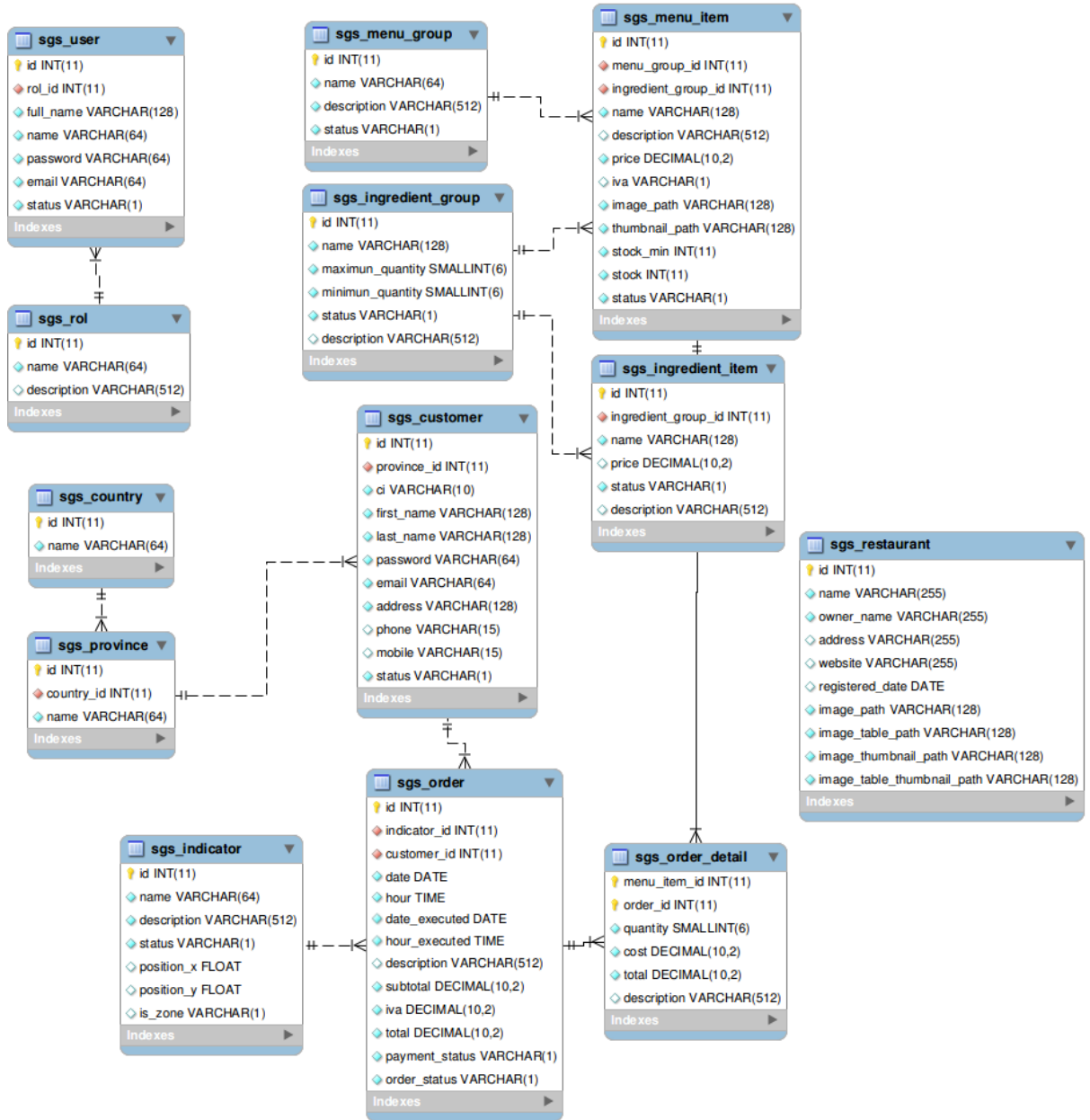


Figura IV - 11 Diagrama de Base de Datos

4.2.9.2.2 Riesgos.

Identificación de los riesgos

TABLA IV - II Identificación de Riesgos

ID	DESCRIPCIÓN DEL RIESGO
R1	El tiempo disponible para la elaboración del proyecto de tesis no es suficiente para concluir satisfactoriamente con el desarrollo de la solución.
R2	Definición ambigua de requerimientos.
R3	Aparición de nuevos requerimientos una vez que el proyecto está en las fases finales.
R4	Reajuste de cronograma de actividades.
R5	Falta de conocimiento de manejo de la herramienta de desarrollo.
R6	El software no satisface con todas las necesidades previstas por el usuario.
R7	Daños en el hardware determinado para el desarrollo del proyecto.

Análisis de los Riesgos

- Criterios de valoración de la probabilidad

TABLA IV - III Criterios de valoración de la probabilidad de riesgos

Rango Probabilidad	Descripción	Valor
1% - 33%	Baja	1
34% - 67%	Media	2
68% - 99%	Alta	3

- Criterios de valoración del impacto

TABLA IV - IV Criterio de valoración del impacto de riesgos

Impacto	Costo	Retraso	Impacto Técnico	Valor
Bajo	<1%	1 semana	Ligero efecto en el desarrollo del proyecto	1
Moderado	<5%	2 semanas	Moderado efecto en el desarrollo del proyecto	2
Alto	<10%	1 mes	Severo efecto en el desarrollo del proyecto	3
Crítico	>10%	Más de 1 mes	Proyecto no puede ser culminado	4

- Determinación de exposición al riesgo

TABLA IV - V Determinación de exposición al riesgo

Exposición al riesgo	Valor	Color
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	>= 6	Rojo

TABLA IV - VI Valoración de determinación de exposición al riesgo

	Bajo = 1	Moderada = 2	Alta = 3	Crítico = 4
Impacto Probabilidad				
Alta = 3	3	6	9	12
Media = 2	2	4	6	8
Baja = 1	1	2	3	4

- Tabla de impactos esperados

TABLA IV - VII Impacto Esperados

Identificación	Probabilidad			Impacto		Exposición al riesgo	
	%	Valor	Probabilidad	Valor	Impacto	Valor	Exposición
R1	50	2	Media	3	Alto	9	Alto
R2	40	2	Media	2	Medio	6	Alto
R3	10	1	Bajo	3	Alto	3	Medio
R4	10	1	Bajo	4	Crítico	4	Medio
R5	15	2	Medio	3	Alto	3	Medio
R6	5	1	Baja	4	Medio	2	Bajo
R7	5	1	Baja	4	Medio	2	Bajo

- Priorización de los Riesgos

TABLA IV - VIII Priorización de los Riesgos

ID	EXPOSICIÓN	
	Valor	Exposición
R1	9	Alto
R2	6	Alto
R3	3	Medio
R4	4	Medio
R5	3	Medio
R6	2	Bajo
R7	2	Bajo

- Plan de Reducción, Supervisión y Gestión del Riesgo

TABLA IV - IX Hoja de Gestión de Riesgo 1

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R1			
Probabilidad: 50	Impacto: Alto	Exposición: Alta	Prioridad: Alta
Valor: 2	Valor: 3	Valor: 9	Valor: 9
Descripción: El tiempo disponible para la elaboración del proyecto de tesis, no es suficiente para concluir satisfactoriamente con el desarrollo de la solución.			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Retraso en la ejecución de las actividades planificadas • Falta de cumplimiento del cronograma • Reducción del tiempo 			
Consecuencias:			
<ul style="list-style-type: none"> • Afecta la calidad del Software • El proyecto no podrá ser entregado en la fecha planificada 			
REDUCCIÓN Y SUPERVISIÓN:			
<ul style="list-style-type: none"> • Elaborar planes para cumplir las fechas planificadas • Adquirir nuevos horarios que permitan lograr el cumplimiento de las actividades 			
SUPERVISIÓN:			
<ul style="list-style-type: none"> • Seguimiento continuo en el desarrollo del proyecto • Incrementar procesos de seguridad en cuanto a la verificación del cronograma 			

GESTIÓN:

Plan de Contingencia:

- Realizar la factibilidad técnica antes del desarrollo del proyecto
- Incrementar procesos de seguridad en cuanto a la verificación del cronograma

ESTADO ACTUAL

Fase de reducción iniciada

Fase de Supervisión iniciada

Gestionando el riesgo

RESPONSABLES:

Karla Carvajal

Christian Sasig

TABLA IV - X Hoja de Gestión de Riesgo 2

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R2			
Probabilidad: 40 Valor: 2	Impacto: Medio Valor: 2	Exposición: Medio Valor: 2	Prioridad: Alta Valor: 6
Descripción: Definición ambigua de requerimientos.			
REFINAMIENTO: Causas: <ul style="list-style-type: none">• Mala comunicación con el cliente• Falta de coordinación entre los miembros del equipo• Acumulación de Información Consecuencias: <ul style="list-style-type: none">• Afecta el desarrollo del sistema• El proyecto no podrá ser entregado en la fecha planificada			
REDUCCIÓN Y SUPERVISIÓN: <ul style="list-style-type: none">• Establecer acuerdos entre los miembros del equipo que lleven al mejoramiento del desarrollo del sistema.• Reorganización de información			
SUPERVISIÓN: <ul style="list-style-type: none">• Seguimiento continuo en el cumplimiento de requerimientos• Evitar modificaciones frecuentes de los requerimientos establecidos.			
GESTIÓN: Plan de Contingencia: <ul style="list-style-type: none">• Establecer claramente los requerimientos con el usuario• Establecer las limitaciones del proyecto de tesis.			
ESTADO ACTUAL Fase de reducción iniciada <input checked="" type="checkbox"/> Fase de Supervisión iniciada <input type="checkbox"/> Gestionando el riesgo <input type="checkbox"/>			
RESPONSABLES: Karla Carvajal Christian Sasig			

4.2.9.3. Codificación

Cliente Siempre Presente

La idea de tener al cliente, un representante de éste o a un usuario no es fácil de asimilar si se consideran los costos que esto representa. En el caso de este proyecto, el cliente no podía desplazarse a ninguno de los lugares de trabajo de los desarrolladores dado que debía estar al frente de su negocio, por tal motivo se debió implementar una estrategia de comunicación distinta en la cual los programadores podían llamar vía telefónica al cliente

Es importante contar con un representante del cliente en las instalaciones del equipo de desarrollo. Sin embargo se puede plantear una solución intermedia donde el representante del cliente solo esté presente durante un periodo de tiempo acordado con el grupo de desarrollo por día, que deberá ser aprovechado para disipar cualquier duda que haya surgido durante el resto del tiempo de desarrollo en que no estuvo.

Estándares del código

La estandarización del código fue asumida desde el mismo momento en que se inició la codificación. Debido que el grupo de desarrollo había estado trabajando unido por largo tiempo, ya tenían un esquema de estándares acordados de forma tácita, sin embargo por seguir una disciplina se formalizaron estos en un documento. (Ver Anexo - Estándares)

Se debe resaltar que el programar empleando estándares es una práctica que no solo se recomienda en XP, es una buena práctica que debe ser seguida en cualquier metodología de desarrollo lo que no implica algo muy novedoso en XP.

4.2.9.4. Pruebas

La creación de pruebas fue una experiencia nueva para el equipo de trabajo al ser una de las reglas de la metodología XP que no se había llegado a utilizar en proyectos anteriores.

El carácter obligatorio de la escritura de las pruebas antes del desarrollo de los métodos del sistema implica un proceso de diseño previo. Esto se considera una ventaja ya que se destina tiempo en la construcción de la prueba, pero al realizar la codificación del método, éste resultaba de manera casi inmediata. También se destaca la autonomía que deben tener dichas pruebas a la hora de su ejecución, lo que implicaba la manipulación de la base de datos y la recuperación de su estado inicial al finalizar la prueba.

Según XP, se deben crear todas las pruebas de una clase antes de comenzar a desarrollar los métodos. En la experiencia fue conveniente realizar las pruebas individualmente, debido a que se producían errores al ejecutar todas las pruebas en un solo llamado. Se descubrió que este inconveniente estaba relacionado con la base de datos, servicios web publicados y con los métodos de consumo desde la aplicación android, lo que al principio del proyecto aplicó dificultad al evaluar si un método había pasado o no la prueba.

Una vez que se descubrió este error se empezó a ejecutar las pruebas por grupos en lugar de ejecutarlas todas de una vez. De esta forma no había problemas y se garantizaba que si una prueba fallaba era solo por errores de lógica.

CONCLUSIONES

Al finalizar el análisis comparativo de las plataformas de desarrollo de desarrollo para dispositivos móviles Android y iOS se han obtenido las siguientes conclusiones:

- Se determinó que el lenguaje de programación Java para el desarrollo de aplicaciones en Android presenta a los desarrolladores una curva de aprendizaje de 75% por su sintaxis comprensible, mientras que el lenguaje de programación objective C tiene un 50% en su curva de aprendizaje por su sintaxis compleja.
- La plataforma Android puede ser instalada en varios sistemas operativos por lo que en el análisis se obtuvo un 75% estos sistemas operativos pueden ser propietarios y libres, mientras que la plataforma iOS obtuvo un 25% ya que solo se ejecuta en sistema operativo Mac OS X.
- La interfaz gráfica de usuario en las dos plataformas de desarrollo de aplicaciones Android y iOS presentan elementos bien acabados estéticamente para el uso en interfaces táctiles.
- El proceso de gestión de la base de datos Sqlite en Android y iOS es bastante simple porque existen librerías nativas para el manejo de la base de datos Sqlite en ambas plataformas y esta base de datos utiliza sentencias sql, las mismas que son conocidas por los programadores.
- Del análisis se obtuvo el 100% en el consumo de servicios web en la plataforma Android ya que existen librerías nativas para este proceso, mientras que en la plataforma iOS se obtuvo el 75% ya que se utiliza librerías externas para este proceso y su utilización depende de cómo fueron implementadas.

- Los IDEs Eclipse y Xcode que son utilizados para el desarrollo de aplicaciones para android y iOS respectivamente, tienen herramientas completas para codificación y diseño de las aplicaciones muy eficientes.
- En ambas plataformas existen emuladores muy eficientes para realizar pruebas a las aplicaciones. Para el despliegue de aplicaciones ya en un dispositivo físico en la plataforma Android es muy fácil por ser software libre, mientras que para iOS el despliegue ocupa un poco más de tiempo y hay que pagar por instalar las aplicaciones en el dispositivo.

RECOMENDACIONES

Al finalizar el proyecto de tesis, se han establecido las siguientes recomendaciones:

- Los parámetros a evaluar en un análisis comparativo deben ser seleccionados cuidadosamente, y estos deben estar orientados a características que permitan la comprobación de la hipótesis.
- Es importante analizar los requerimientos del usuario y el entorno para realizar una aplicación en la plataforma Android o iOS
- Se debe valorar los costos y el tiempo para la realización en la una u otra plataforma ya que si se va a desarrollar para iOS suben los costos y en cambio en Android como es una plataforma libre se puede realizar la misma aplicación con menores costos.
- Para desarrollar una aplicación para Android es recomendable utilizar el IDE Eclipse y el plugin Motodev Studio para Android estas dos herramientas ofrecen un conjunto completo de características para el desarrollador.
- Se recomienda a los desarrolladores de aplicaciones que desean utilizar la plataforma iOS familiarizarse con el lenguaje C para mayor comprensión
- Se recomienda realizar aplicaciones utilizando estándares de codificación para una mejor comprensión código.
- Se recomienda ayudarse de la documentación oficial para el desarrollo en las plataformas Android y iOS ya que continuamente es actualizada.
- Se recomienda para realizar las pruebas de aplicaciones hacerlas también en un dispositivo físico ya que los emuladores tienen limitaciones.

RESUMEN

Análisis comparativo entre las plataformas de desarrollo de aplicaciones móviles para los sistemas operativos android y ios, aplicado al Restaurante Bonny de la ciudad de Riobamba.

Se utilizó el método científico con las técnicas de observación y técnicas de estudios de caso en la creación de prototipos para ambas plataformas de desarrollo que permitieron dar resultados de manera cuantitativa y cualitativa a la investigación. Se utilizó los siguientes materiales, en elementos hardware: 2 computadores, 1 tablet y en elementos software: Eclipse, SDK de android, Netbeans, Mysql, Symfony Framework.

Mediante los parámetros de medición, se obtuvo los siguientes puntajes: Android 93,75% y iOS con 77,5%, siendo el de mayor productividad en la implementación de aplicaciones la plataforma Android con una equivalencia de Muy bueno. Implementando el sistema de recolección de órdenes en el restaurante con este último en una aplicación Nativa para Android, el mismo que cuenta con un diseño de ambientes amigables y es de fácil uso para el usuario.

Se concluye que la plataforma de desarrollo Android mejora la productividad en la implementación de aplicaciones porque en el desarrollo de la mayor parte de los parámetros establecidos obtuvo un mayor desempeño.

Se recomienda para la realización de proyectos en aplicaciones móviles utilizar la plataforma Android ya que ofrece un conjunto de herramientas que facilitan la implementación de aplicaciones, brinda una extensa documentación y existe una gran comunidad de desarrolladores.

ABSTRACT

“Comparative analysis mobile application development platforms for Android and IOS operative systems”

The market has growth in intelligent mobile devices. This promotes designers and companies to develop applications and services for these devices, and they can choose one platform to obtain the maximum productivity in the implementation of these applications.

The objectives are the following:

1. To study the development platforms for intelligent mobile devices.
2. To analyze the application development platform for android and ios operative systems.
3. To determine the comparisons to obtain a bigger productivity in applications development for android and iOS.
4. To select the development platform based on established comparison parameters and prototypes' tests.
5. To design a software application based on selected platform for customer service in Bonny restaurant' orders.

The research used the scientific method with observation and study techniques for the creation of prototypes for both development platforms which allow quantitative and qualitative results. The materials were the following. Hardware elements: 2 computers and 1 tablet. Software elements: Eclipse, SDK from Android, Netbeans, Mysql, Symfony Framework.

With measurement parameters, the research obtained these results: Android 93.75% and iOS 77.5%. Android has the best productivity with the platform application implemented with an equivalence of “Very good”. This way the

restaurant counts with a system of orders collection with a native Android application, and it counts with a user friendly design and environment.

Android improves productivity in application implementing because of a great performance in most of the established parameters.

There be done projects with mobile applications with Android which uses a group of tools to make the application implementing easy with a large documentation and community of developers.

ANEXOS

Anexo 1. Historias de usuario

Se han definido las siguientes historias de usuario:

TABLA IV - XI Historia de Usuario 1

Descripción del proceso de negocio	
Número: 1	Usuario: Administrador
Nombre proceso: Autenticación de los usuarios	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Puntos Estimados: 3	Interacción asignada: 2
Programador Responsable: Karla Carvajal, Christian Sasig	
Todos los usuarios deben autenticarse antes de usar el sistema, este proceso permite ingresar a las opciones propias de cada usuario. Los usuarios existentes en el sistema son: <ul style="list-style-type: none">- Administrador- Usuarios(Meseros)	

TABLA IV - XII Historia de Usuario 2

Descripción del proceso de negocio	
Número: 2	Usuario: Administrador
Nombre proceso: Crear Usuarios	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Puntos Estimados: 3	Interacción asignada: 2
Programador Responsable: Karla Carvajal, Christian Sasig	
En primera instancia el Administrador debe iniciar sesión con su nombre de usuario y contraseña, a continuación le permitirá visualizar todos los procesos disponibles para este usuario, siendo el Administrador el único responsable de la creación de usuarios. Una vez ingresado al proceso de creación el administrador deberá llenar todos los campos con información del usuario nuevo y confirmar la creación.	

TABLA IV - XIII Historia de Usuario 3

Descripción del proceso de negocio	
Número: 3	Usuario: Administrador
Nombre proceso: Crear Categorías de Platos	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Puntos Estimados: 3	Interacción asignada: 2
Programador Responsable: Karla Carvajal, Christian Sasig	
Una vez que existan usuarios disponibles se procederá a crear las categorías de platos, el responsable de este proceso es el Administrador, el mismo que debe estar previamente autenticado.	

TABLA IV - XIV Historia de Usuario 4

Descripción del proceso de negocio	
Número: 4	Usuario: Jefe de Proyecto
Nombre proceso: Crear Platos	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Puntos Estimados: 3	Interacción asignada: 2
Programador Responsable: Karla Carvajal, Christian Sasig	
Una vez creadas las categorías de platos. El Administrador debe ingresar los platos escogiendo la categoría la que pertenece.	

TABLA IV - XV Historia de Usuario 5

Descripción del proceso de negocio	
Número: 5	Usuario: Jefe de Proyecto
Nombre proceso: Crear Orden	
Prioridad en negocio: Alta	Riesgo de desarrollo: Medio
Puntos Estimados: 3	Interacción asignada: 2
Programador Responsable: Karla Carvajal, Christian Sasig	
El Administrador o los usuarios deberán ingresar al sistema con su nombre de usuario y su contraseña. Una vez en el menú propio del usuario, podrá navegar por los diferentes platos, agregarlos a su orden y enviar la orden a la cocina.	

Anexo 2. Casos de Uso
Descripción de los Procesos en la Aplicación Servidor

TABLA IV - XVI Caso de Uso Autenticar

Identificador Caso de Uso	Caso de Uso Autenticar
Nombre del Caso de Uso	CU_Autenticar
Actores	Administrador, Usuarios
Propósito	Verificar si un usuario está registrado en el sistema.
Visión General	El sistema verifica si un usuario está registrado dentro del sistema y le permite o niega el acceso de la aplicación
Tipo	Esencial y expandida.
Referencias	
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. Iniciar cuando el usuario requiere acceder a la aplicación	2. Muestra una ventana para que el usuario ingrese su nombre de usuario y password
3. El usuario ingresa su nombre de usuario y password	4. Muestra una ventana de la aplicación
5. El usuario utiliza la aplicación	
Cursos Alternos	
Línea 4: Nombre de usuario y/o password son incorrectas, se indica el error	

TABLA IV - XVII Caso de Uso Cerrar Sesión

Identificador Caso de Uso	Caso de Uso Cerrar Sesión
Nombre del Caso de Uso	CU_Cerrar_Sesión
Actores	Administrador, Usuarios
Propósito	Permitir que un usuario que está utilizando la aplicación pueda cerrar su sesión
Visión General	El usuario al momento que desee finalizar su sesión correspondiente, después de haber realizado todas sus actividades a las que tiene acceso dentro del sistema, procederá a cerrar la sesión.
Tipo	Esencial y expandido.
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El usuario cierra su sesión	2. El sistema cierra la sesión activa y presenta una página para un nuevo inicio de sesión.
Cursos Alternos	
Línea 1: Si el usuario no cierra la sesión correctamente esto puede ser una falta grave de seguridad.	

TABLA IV - XVIII Caso de Uso Agregar Categoría de platos

Identificador Caso de Uso	Caso de Uso Agregar Categoría de Platos
Nombre del Caso de Uso	CU_Agregar_Categoria_Platos
Actores	Administrador
Propósito	Agregar categorías de platos a la aplicación
Visión General	El Administrador es el encargado de agregar nuevas categorías.
Tipo	Esencial, expandido
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Categorías de Productos.	4. Ingresa los datos de la categoría y su descripción
5. Guarda los datos.	6. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5 y 6: Si los requerimientos no están completos muestra un mensaje de error	

TABLA IV - XIX Caso de Uso Agregar Categorías de Ingredientes

Identificador Caso de Uso	Caso de Uso Agregar Categoría de Ingredientes
Nombre del Caso de Uso	CU_Agregar_Categoria_Ingredientes
Actores	Administrador
Propósito	Agregar categorías de ingredientes a la aplicación
Visión General	El Administrador es el encargado de agregar nuevas categorías.
Tipo	Esencial, expandido
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Categorías de Ingredientes.	4. Ingresa los datos de la categoría y su descripción
5. Guarda los datos.	6. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5 y 6: Si los requerimientos no están completos muestra un mensaje de error	

TABLA IV - XX Caso de Uso Agregar Ingrediente

Identificador Caso de Uso	Caso de Uso Agregar Ingrediente
Nombre del Caso de Uso	CU_Agregar_Ingrediente
Actores	Administrador
Propósito	Agregar ingredientes a la aplicación
Visión General	El Administrador es el encargado de agregar nuevos ingredientes.
Tipo	Esencial, expandido
Referencias	CU_Autenticar, CU_Agregar_Categoria_Ingredientes
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Ingredientes.	4. Ingresa los datos del ingrediente y su descripción
5. Guarda los datos.	6. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5 y 6: Si los requerimientos no están completos muestra un mensaje de error	

TABLA IV - XXI Caso de Uso Agregar Plato

Identificador Caso de Uso	Caso de Uso Agregar Plato
Nombre del Caso de Uso	CU_Agregar_Plato
Actores	Administrador
Propósito	Agregar platos a la aplicación
Visión General	El Administrador es el encargado de agregar nuevos platos.
Tipo	Esencial, expandido
Referencias	CU_Autenticar, CU_Agregar_Categoria_Platos
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Platos.	4. Ingresa los datos del plato y su descripción
5. Guarda los datos.	6. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5 y 6: Si los requerimientos no están completos muestra un mensaje de error	

TABLA IV - XXII Caso de Uso Agregar Mesa

Identificador Caso de Uso	Caso de Uso Agregar Mesa
Nombre del Caso de Uso	CU_Agregar_Mesa
Actores	Administrador
Propósito	Agregar platos a la aplicación
Visión General	El Administrador es el encargado de agregar nuevas mesas.
Tipo	Esencial, expandido
Referencias	CU_Autenticar,
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Mesas.	4. Ingresa los datos de la mesa y su descripción
5. Guarda los datos.	6. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5 y 6: Si los requerimientos no están completos muestra un mensaje de error	

TABLA IV - XXIII Caso de Uso Listar Ordenes del día

Identificador Caso de Uso	Caso de Uso Listar órdenes del día
Nombre del Caso de Uso	CU_Listar_Ordenes_día
Actores	Administrador, Usuarios
Propósito	Listar las órdenes del día
Visión General	El Administrador o los usuarios son los encargados de listar las órdenes del día.
Tipo	Esencial, expandido
Referencias	CU_Autenticar,
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Gestión de Ordenes-Ordenes del día.	4. Visualiza listado de órdenes del día.
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	

TABLA IV - XXIV Caso de Uso Ejecutar Orden

Identificador Caso de Uso	Caso de Uso Ejecutar_Orden
Nombre del Caso de Uso	CU_Ejecutar_Orden
Actores	Administrador, Usuario
Propósito	Cambiar el estado a ejecutado cuando la orden salga de la cocina.
Visión General	Una vez preparada la orden en la cocina, El administrador o los usuarios cambiaran el estado de la orden.
Tipo	Esencial, expandido
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El miembro del equipo debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del jefe de equipo.
3. Ingresa al módulo órdenes	4. Visualiza las órdenes del día.
5. Selecciona la orden.	6. Cambiar el estado de la orden a ejecutada.
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	

TABLA IV - XXV Caso de Uso Agregar Usuarios

Identificador Caso de Uso	Caso de Uso Agregar_Usuario
Nombre del Caso de Uso	CU_Agregar_Usuario
Actores	Administrador
Propósito	Agregar usuarios al Sistema
Visión General	El Administrador es el encargado de agregar nuevos Usuarios al sistema
Tipo	Esencial, expandido
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades del administrador
3. Ingresa al módulo Usuarios.	4. Ingresa los datos del usuario.
5. Asignar el rol que tiene el usuario.	
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4 y 5: Si los requerimientos presentados no están completos se visualiza un mensaje de error	

TABLA IV - XXVI Caso de uso Generar Reporte

Identificador Proceso	Proceso Generar_Reporte
Nombre del Proceso	Generar Reporte
Actores	Administrador
Propósito	Permitir generar reporte
Visión General	El Administrador solicita generar el reporte.
Tipo	Esencial, expandido
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla vinculo del reporte.
3. Solicitar la generación del reporte	4. Presenta un reporte con la información solicitada
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	

Descripción de los Procesos en la Aplicación Cliente

TABLA IV - XXVII Caso de Uso Autenticar

Identificador Caso de Uso	Caso de Uso Autenticar
Nombre del Caso de Uso	CU_Autenticar
Actores	Administrador, Usuarios
Propósito	Verificar si un usuario está registrado en el sistema.
Visión General	El sistema verifica si un usuario está registrado dentro del sistema y le permite o niega el acceso de la aplicación
Tipo	Esencial y expandida.
Referencias	
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. Iniciar cuando el usuario requiere acceder a la aplicación	2. Muestra una ventana para que el usuario ingrese la dirección ip, su nombre de usuario y password
3. El usuario ingresa la dirección ip ,su nombre de usuario y password	4. Muestra una ventana de la aplicación
5. El usuario utiliza la aplicación	
Cursos Alternos	
Línea 4: Nombre de usuario y/o password son incorrectas, se indica el error	

TABLA IV - XXVIII Caso de Uso Cerrar Sesión

Identificador Caso de Uso	Caso de Uso Cerrar Sesión
Nombre del Caso de Uso	CU_Cerrar_Sesión
Actores	Administrador, Usuarios
Propósito	Permitir que un usuario que está utilizando la aplicación pueda cerrar su sesión
Visión General	El usuario al momento que desee finalizar su sesión correspondiente, después de haber realizado todas sus actividades a las que tiene acceso dentro del sistema, procederá a cerrar la sesión.
Tipo	Esencial y expandido.
Referencias	CU_Autenticar
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El usuario cierra su sesión	2. El sistema cierra la sesión activa y presenta una página para un nuevo inicio de sesión.
Cursos Alternos	
Línea 1: Si el usuario no cierra la sesión correctamente esto puede ser una falta grave de seguridad.	

TABLA IV - XXIX Caso de Uso Seleccionar la Mesa

Identificador Caso de Uso	Caso de Uso Seleccionar la mesa
Nombre del Caso de Uso	CU_Seleccionar_Mesa
Actores	Administrador, Usuarios
Propósito	Seleccionar la mesa
Visión General	El Administrador o los usuarios son los encargados de seleccionar la mesa.
Tipo	Esencial, expandido
Referencias	CU_Autenticar,
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades
3. Ingresa al módulo de mesas.	4. Visualiza las mesas disponibles
5. Selecciona la mesa	6. Pasa a la siguiente actividad
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	

TABLA IV - XXX Caso de Uso Agregar Orden

Identificador Caso de Uso	Caso de Uso Agregar Orden
Nombre del Caso de Uso	CU_Agregar_Orden
Actores	Administrador, Usuarios
Propósito	Agregar órdenes a la aplicación
Visión General	El Administrador o los usuarios son los encargados de agregar nuevas órdenes.
Tipo	Esencial, expandido
Referencias	CU_Autenticar, CU_Seleccionar_Mesa
Curso Típico de Eventos	
Acción de los Actores	Respuesta del Sistema
1. El Administrador debe autenticarse con su Nombre de usuario y password	2. Visualiza pantalla de actividades
3. Ingresa al módulo de platos.	4. Visualiza las categorías de platos
5. Selecciona los platos y selecciona ordenar	6. Muestra un mensaje de confirmacion
7. Confirma el envío de la orden	8. Muestra una notificación
Cursos Alternos	
Línea 1: Si el usuario o password no son correctos se visualiza mensaje de error	
Línea 4,5,6,7 y 8: Si los requerimientos no están completos muestra un mensaje de error	

Anexo 3. Estándares

Los siguientes estándares responden a una serie de buenas prácticas recomendadas para desarrollar aplicaciones y promovidas entre otros por los grupos de usuarios de JAVA.

Si bien no se adoptaron todos los estándares recomendados para desarrollar en JAVA y se implementaron otros nuevos, lo que se buscó fue facilitar la comprensión del código para a su vez lograr la propiedad colectiva del código.

- Todo paquete inicia con minúscula.
- Toda clase inicia con mayúscula.
- Todo método inicia con minúscula.

- Toda variable inicia con minúscula.

Sea clase, método o variable. Si consta de dos o más palabras. Todas van sin espacios y de la segunda en adelante con mayúscula inicial.

Crear JavaDoc: Antes de todo método, variable y clase debe ir una explicación del mismo empleando el siguiente formato, con excepción de aquellos en que su comprensión resulte intuitiva a juicio del desarrollador, acogiéndose a la simplicidad según lo plantea XP.

```
o /** comentario */  
o /**  
* comentario  
*/
```

Todo el código debe estar correctamente estructurado.

- Se abre una llave en la siguiente línea vacía.
- En la misma columna de la primera letra de la línea anterior.
- No se escribe nada más en esa línea.
- Se cierra en la misma columna donde fue abierto.
- Tampoco se escribe nada en la línea donde se cerró.
- EXCEPCIONES
- Cuando no hay cuerpo dentro de las llaves o solo hay una línea.
- Los try-catch.

```
try{  
  CUERPO  
}catch(Exception){  
  CUERPO  
}
```

- Solo se declaran variables al inicio de una clase, método o ciclo.
- La hora en formato de 24 horas.
- El nombre de las tablas es en singular y minúscula.
- Los objetos que administran la lógica del negocio tienen el mismo nombre que la tabla símil en la base de datos respetando los estándares anteriores.

- Las columnas de una tabla se convierten en atributos privados en la respectiva clase manteniendo el tipo de dato.

GLOSARIO

ADT: Herramienta de desarrollo de Android (ADT) es un plugin para el IDE Eclipse que está diseñado para darle un ambiente potente, integrado en el que se puedan crear aplicaciones para Android.

Android: Es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

App Store: Es un servicio para el iPhone, el iPod Touch, el iPad, Mac OS X Snow Leopard y Mac OS X Lion, creado por Apple Inc., que permite a los usuarios buscar y descargar aplicaciones informáticas de iTunes Store o Mac App Store en el caso de Mac OSX, desarrolladas con el iPhone SDK y publicadas por Apple.

Cocoa Touch: Es un API para la creación de programas para el iPad, iPhone y iPod Touch de la compañía Apple Inc. Cocoa Touch proporciona una capa de abstracción al sistema operativo iOS.

Dalvik Virtual Machine: Dalvik es la máquina virtual que utiliza la plataforma para dispositivos móviles Android. Dalvik está optimizada para requerir poca memoria y está diseñada para permitir ejecutar varias instancias de la máquina virtual simultáneamente.

DDMS: Es un monitor de depuración de la máquina virtual Dalvik, que es la máquina virtual de proceso de Android. Esta aplicación proporciona servicios de captura de pantallas, captura y listado de información en el dispositivo, hacer logcats, ver los procesos y simular llamadas entrantes o SMS, e incluso simular una posición de ubicación.

iOS: Anteriormente denominado iPhone OS es un sistema operativo móvil de Apple. Originalmente desarrollado para el iPhone, siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros.

Google Play: Antes llamado Android Market es una tienda de software en línea desarrollada por Google para los dispositivos Android. Es una aplicación ("app") que está preinstalada en la mayoría de los dispositivos Android y que permite a los usuarios buscar y descargar aplicaciones publicadas por desarrolladores terceros, alojada en Google Play.

Xcode: Es el entorno de desarrollo integrado (IDE) de Apple Inc. y se suministra gratuitamente junto con Mac OS X. Xcode trabaja conjuntamente con Interface Builder una herramienta gráfica para la creación de interfaces de usuario.

SQLite: Es un sistema de gestión de bases de datos relacional contenida en una pequeña biblioteca realizada en C esta librería no es un proceso independiente sino que se enlaza con el programa que la utiliza.

BIBLIOGRAFIA

- 1.- **HERNANDEZ, S.**, Metodología de la investigación., 6a. ed., México Distrito Federal – México., McGRAW., 1997., Pp-297.
- 2.- **GARGENTA, M.**, Learning Android., Nueva York–Estados Unidos., O’Reilly., 2011., Pp: 260- 280.
- 3.- **LEE, W.**, Beginning iOS 4 Application Development., Ottawa– Canada., Wrox., 2011., Pp: 675- 690.
- 4.- **LEWIS, R.**, Aplicaciones iPhone e iPad para principiantes Android., Nueva York–Estados Unidos., O’Reilly., 2011., Pp: 260- 280.
- 5.- **ROCHE, K.**, Benning Java Google App Engine., Nueva York–Estados Unidos., O’Reilly., 2009., Pp: 330- 420.

BIBLIOGRAFÍA DE INTERNET

- 6.- Bases de datos en android

<http://www.sgoliver.net/blog/?p=1611>

2012-01-10

- 7.- Creando tareas asíncronas en android con AsyncTask

<http://miguelangellv.wordpress.com/2011/01/31/creando-tareas-asincronas-en-android-con-async-task/>

2011-11-10

- 8.- Documentación del sdk de Android

<http://developer.android.com/index.html>

2011-10-22

9.- Documentación del sdk de iOS

<https://developer.apple.com/devcenter/ios/index.action>

2011-10-22

10.- Documentación de Symfony 2

<http://symfony.com/>

2011-11-10

11.- Ejemplo de sqlite en iOS

http://www.techotopia.com/index.php/An_Example_SQLite_based_iOS_4_iPhone_Application

2012-01-10

12.- Protocolo SOAP en iOS

<http://www.iphonesoftware.es/2010/06/10/protocolo-soap-en-ios/>

2012-01-10