

# UNIVERSIDAD DE BURGOS

## ESCUELA POLITÉCNICA SUPERIOR



### Ingeniería Técnica en Informática de Gestión

### **Android Server Status: Aplicación Android para la gestión remota de servidores**

Proyecto Final de Carrera

Alumnos      David Herrero de la Peña  
                  Jonatan Santos Barrios

Tutor      Carlos López Nozal  
                  DEPARTAMENTO DE INGENIERÍA CIVIL  
                  Área de Lenguajes y Sistemas Informáticos  
                  Rubén Arribas Barrio

Burgos, 26 de junio de 2012



---

# Índice de contenido

<b>Parte I: Descripción del proyecto.....</b>	<b>15</b>
1. Introducción.....	20
1.1 .Estructura de la memoria.....	20
2. Objetivos del proyecto.....	21
2.1 .Objetivos principales.....	21
2.2 .Objetivos técnicos.....	21
3. Conceptos teóricos.....	22
3.1 .Gestión de servidores.....	22
3.2 .Web Service.....	22
3.3 .GPS.....	23
3.4 .Syslog.....	24
3.5 .Estilos de diseño de aplicaciones Android.....	24
3.6 .Scrum.....	25
3.7 .Sistema de control de versiones.....	26
3.8 .Sistema de control de tareas.....	26
4. Técnicas y herramientas.....	27
4.1 .Lenguaje Java.....	27
4.2 .Android SDK.....	28
4.3 .SQLite.....	28
4.4 .MySQL.....	29
4.5 .Apache Axis2.....	29
4.6 .Apache Tomcat.....	29
4.7 .JSVC.....	30
4.8 .JSCH.....	30
4.9 .Eclipse.....	30
4.10 .Git.....	31
4.11 .GitHub.....	31
4.12 .TortoiseGIT.....	31
4.13 .Open Office.....	32
4.14 .Astah Community.....	32
4.15 .JMeter.....	32
4.16 .Droid Explorer.....	32
5. Aspectos relevantes del desarrollo.....	32
5.1 .Versiones de Android.....	32
6. Trabajos relacionados.....	33
7. Conclusiones y lineas de trabajo futuras.....	34
7.1 .Conclusiones.....	34
7.2 .Lineas de trabajo futuras.....	35
<b>Anexo I: Plan del proyecto software.....</b>	<b>37</b>
1. Introducción.....	42
2. Planificación temporal.....	42
2.1 .Planificación incial.....	42
2.1.1 .Fase de análisis.....	42

2.1.2 .Fase de diseño.....	43
2.1.3 .Fase de implementación.....	43
2.1.4 .Fase de pruebas.....	43
2.1.5 .Fase de documentación.....	43
2.1.6 .Diagrama de Gantt de la planificación inicial.....	43
2.1.7 .Hoja de tareas de la planificación inicial.....	45
2.2 .Planificación final.....	45
2.2.1 .Diagrama de Gant de la planificación final.....	46
2.2.2 .Hoja de tareas de la planificación final.....	47
3. Estudio de viabilidad.....	47
3.1 .Viabilidad económica.....	47
3.1.1 .Costes de personal.....	48
3.1.2 .Costes de hardware.....	49
3.1.3 .Costes de software.....	49
3.1.4 .Costes generales.....	51
3.1.5 .Costes totales.....	51
3.1.6 .Análisis de costes-beneficios.....	52
3.2 .Viabilidad legal.....	52
<b>Anexo II: Especificación de requisitos.....</b>	<b>53</b>
1. Introducción.....	58
2. Objetivos del proyecto.....	58
3. Lista de usuarios participantes.....	58
4. Descripción del sistema actual.....	58
5. Catalogo de requisitos del sistema.....	59
5.1 .Requisitos funcionales.....	59
5.1.1 .Requisitos funcionales módulo control de accesos.....	59
5.1.2 .Requisitos funcionales módulo teminal SSH.....	61
5.1.3 .Requisitos funcionales módulo localización GPS.....	62
5.2 .Requisitos de información.....	63
5.2.1 .Web Service.....	63
5.2.2 .Módulo terminal SSH.....	64
5.2.3 .Módulo control de accesos.....	65
5.2.4 .Módulo localizador GPS.....	67
5.3 .Requisitos no funcionales.....	67
6. Especificaciones de requisitos.....	68
6.1 .Diagrama de casos de uso.....	68
6.2 .Casos de uso.....	69
7. Interfaz de usuario.....	77
8. Pruebas.....	80
<b>Anexo III: Especificaciones de diseño.....</b>	<b>83</b>
1. Introducción.....	91
2. Ámbito de software.....	91
3. Diseño de datos.....	92
3.1 .Objetos de datos y estructuras de datos.....	92
3.1.1 .Módulo Consola SSH.....	92

---

3.1.2 .Módulo Control de Accesos.....	93
3.2 .Estructuras de archivos y bases de datos.....	93
3.2.1 .Web Service.....	95
3.2.1.1 .Modelo lógico.....	96
3.2.1.2 .Modelo físico.....	96
3.2.2 .Módulo Consola.....	96
3.2.2.1 .Modelo lógico.....	96
3.2.2.2 .Modelo físico.....	97
3.2.3 .Módulo GPS.....	97
3.2.3.1 .Modelo lógico.....	97
3.2.3.2 .Modelo físico.....	98
3.2.4 .Módulo Control de Accesos.....	98
3.2.4.1 .Modelo lógico.....	98
3.2.4.2 .Modelo físico.....	98
4. Diseño arquitectónico.....	99
4.1 .Estilos arquitectónicos utilizados.....	99
4.1.1 .Cliente-Servidor.....	99
4.1.2 .Arquitectura en capas.....	100
4.2 .Patrones de diseño utilizados.....	101
4.2.1 .Singleton.....	101
4.2.2 .Fachada.....	101
4.3 .Diseño general.....	102
4.4 .Descripción detallada.....	103
4.4.1 .Módulo Consola.....	104
4.4.2 .Módulo GPS.....	105
4.4.3 .Módulo Control de Accesos.....	106
4.4.4 .Principal.....	107
4.4.5 .Web Services.....	107
4.5 .Diagramas de clases.....	107
4.5.1 .Módulo Consola.....	107
4.5.2 .Módulo GPS.....	111
4.5.3 .Módulo Control de Accesos.....	114
4.5.4 .Principal.....	118
4.5.5 .Web Service.....	118
5. Diseño de la interfaz.....	120
5.1 .Principal.....	120
5.1.1 .Pantalla de login.....	121
5.1.2 .Pantalla launcher.....	122
5.1.3 .Pantalla falsa.....	123
5.2 .Módulo Consola.....	123
5.2.1 .Pantalla principal.....	124
5.2.2 .Pestaña manual.....	125
5.2.3 .Pestaña favoritos.....	126
5.2.4 .Formulario.....	127
5.2.5 .Terminal SSH.....	128
5.2.6 .Pantalla comandos ejecutados.....	129
5.2.7 .Pantalla comandos favoritos.....	130

5.3 .Módulo GPS.....	131
5.3.1 .Pantalla principal.....	131
5.4 .Módulo Control de Accesos.....	132
5.4.1 .Filtro.....	132
5.4.2 .Formulario.....	133
5.4.3 .Main.....	134
5.4.4 .Pestaña favoritos.....	135
5.4.4.1 .Lista servidores.....	136
5.4.5 .Pestaña notificaciones.....	137
6. Diseño procedimental.....	137
6.1 .Principal.....	138
6.1.1 .Login.....	138
6.1.2 .Main.....	139
6.2 .Módulo SSH.....	140
6.2.1 .Guardar nuevo servidor.....	140
6.2.2 .Conectar a servidor.....	140
6.2.3 .Enviar comando.....	142
6.2.4 .Guardar script.....	143
6.2.5 .Ejecutar script guardado.....	144
6.3 .Módulo GPS.....	145
6.3.1 .Iniciar desde sms.....	145
6.3.2 .Guardar localizaciones.....	146
6.3.3 .Enviar correo electrónico.....	147
6.3.4 .Enviar sms.....	148
6.3.5 .Autorranque al encender.....	149
6.4 .Módulo Consola de Acessos.....	149
6.4.1 .Añadir servidor SSH.....	149
6.4.2 .Monitorizar servidor SSH.....	149
6.4.3 .Descargar notificaciones nuevas y guardadas.....	150
6.4.4 .Filtrar notificaciones.....	152
6.4.5 .Eliminar filtro.....	154
6.4.6 .Borrar notifcaciones.....	154
6.5 .Web Service.....	156
6.5.1 .Borrar notificación.....	156
6.5.2 .Hay notificaciones.....	156
6.5.3 .Obtener notificaciones.....	156
6.5.4 .ScanLog.....	157
7. Referencias cruzadas a los requisitos.....	157
8. Pruebas.....	158
8.1 .Pruebas de stress.....	158
8.2 .Pruebas de integración.....	159
8.3 .Tablas de pruebas.....	159
8.3.1 .Principal.....	159
8.3.2 .Módulo SSH.....	160
8.3.2.1 .AccesoDatos.....	160
8.3.2.2 .Vista.....	161
8.3.3 .Módulo GPS.....	161

---

8.3.3.1 .AccesoDatos.....	161
8.3.3.2 .Vista.....	162
8.3.4 .Módulo Control de Accesos.....	163
8.3.4.1 .Vista.....	163
8.3.5 .Web Service.....	163
9. Entorno tecnológico del sistema.....	163
9.1 .Restricciones técnicas.....	164
9.2 .Comunicación dentro del sistema.....	165
<b>Anexo IV: Documentación técnica de programación.....</b>	<b>167</b>
1. Introducción.....	172
2. Documentación de las bibliotecas.....	172
2.1 .MySQLJDBC.....	172
2.2 .Jsvc.....	172
2.3 .Jsch.....	172
3. Manual del programador.....	172
3.1 .Instalación Java JDK.....	173
3.1.1 .Instalación en Windows.....	173
3.1.2 .Instalación en Linux (Ubuntu).....	173
3.2 .Instalación Android SDK.....	174
3.3 .Instalación Eclipse.....	175
3.3.1 .Instalación plugin ADT.....	176
3.3.2 .Instalación librerías externas.....	176
3.4 .Instalación Apache Tomcat 7.0.....	177
3.5 .Instalación Apache Axis2.....	179
3.6 .Instalación MySQL.....	180
3.6.1 .Instalación conector MySQLJDBC.....	181
3.7 .Configuración Syslog.....	181
3.8 .Instalación Jsvc.....	182
4. Manual de compilación.....	182
4.1 .Compilación de aplicaciones para Android.....	182
4.2 .Compilación del Web Service.....	184
4.3 .Compilación del Demonio.....	193
5. Pruebas.....	197
5.1 .Pruebas de integración.....	197
5.1.1 .Pruebas sobre el Launcher.....	197
5.1.2 .Pruebas sobre el Terminal SSH.....	197
5.1.3 .Pruebas sobre el Control de Accesos.....	198
5.1.4 .Pruebas sobre el localizador GPS.....	199
5.2 .Pruebas de stress.....	199
<b>Anexo V: Manuales de usuario.....</b>	<b>,205</b>
1. Introducción.....	210
2. Manual de instalación.....	210
2.1 .Instalación aplicaciones Android.....	210
2.2 .Instalación Web Service.....	210
2.3 .Instalación Demonio.....	211

3. Manual de usuario.....	212
3.1 .Web Service.....	213
3.2 .Demonio.....	213
3.3 .Cliente Android.....	214
3.3.1 .Launcher.....	214
3.3.2 .Terminal SSH.....	218
3.3.3 .Localizador GPS.....	222
3.3.4 .Control de accesos.....	225

---

## Índice de ilustraciones

Ilustración 1: Comunicación Web Service [5].....	23
Ilustración 2: Diagrama de la metodología Scrum [13].....	26
Ilustración 3: Distribución de las versiones de Android en Junio de 2012 [30].....	33
Ilustración 4: Diagrama de Gantt de la planificación inicial.....	44
Ilustración 5: Tareas de la planificación inicial.....	45
Ilustración 6: Diagrama de Gantt de la planificación final.....	46
Ilustración 7: Tareas de la planificación final.....	47
Ilustración 8: Módelo Entidad-Relación Web Service.....	64
Ilustración 9: Módelo Entidad-Relación Scripts.....	64
Ilustración 10: Módelo Entidad-Relación Servidores.....	65
Ilustración 11: Módelo Entidad-Relación Email.....	66
Ilustración 12: Módelo Entidad-Relación Servidores.....	66
Ilustración 13: Módelo Entidad-Relación Coordenadas.....	67
Ilustración 14: Diagrama de casos de uso.....	68
Ilustración 15: Diagrama de secuencia de una conexión SSH.....	71
Ilustración 16: Diagrama de secuencia de activación del localizador GPS.....	73
Ilustración 17: Diagrama de secuencia del control de accesos.....	75
Ilustración 18: Pantalla principal.....	78
Ilustración 19: Pantalla de opciones.....	79
Ilustración 20: Pantalla con pestañas.....	80
Ilustración 21: Fichero xml.....	94
Ilustración 22: Cliente/Servidor.....	99
Ilustración 23: Capas.....	100
Ilustración 24: Singleton Conexion.....	101
Ilustración 25: Fachada.....	102
Ilustración 26: Diagrama de paquetes.....	103
Ilustración 27: Paquetes.....	104
Ilustración 28: modulossch accesodatos.....	108
Ilustración 29: modulossch modelo.....	109
Ilustración 30: modulossch logica.....	110
Ilustración 31: modulogps accesodatos.....	111
Ilustración 32: modulogps modelo.....	112
Ilustración 33: modulogps logica.....	113
Ilustración 34: modulocontrol accesodatos.....	114
Ilustración 35: modulocontrol modelo.....	115
Ilustración 36: modulocontrol logica1.....	116
Ilustración 37: modulocontrol logica2.....	117
Ilustración 38: Principal.....	118
Ilustración 39: Web Service analizador.....	119
Ilustración 40: Web Service notificador.....	119
Ilustración 41: Principal login.....	121
Ilustración 42: Principal launcher.....	122
Ilustración 43: Principal falsa.....	123
Ilustración 44: ssh principal.....	124

Ilustración 45: ssh pestaña manual.....	125
Ilustración 46: ssh servidores favoritos.....	126
Ilustración 47: ssh formulario.....	127
Ilustración 48: ssh consola.....	128
Ilustración 49: ssh comandos ejecutados.....	129
Ilustración 50: ssh comandos favoritos.....	130
Ilustración 51: gps pantalla principal.....	131
Ilustración 52: control filtro.....	132
Ilustración 53: control formulario.....	133
Ilustración 54: control main.....	134
Ilustración 55: control pestaña.....	135
Ilustración 56: control lista servidores.....	136
Ilustración 57: control pestaña notificaciones.....	137
Ilustración 58: Secuencia principal login.....	138
Ilustración 59: principal secuencia main.....	139
Ilustración 60: ssh nuevo servidor.....	140
Ilustración 61: ssh conectar.....	141
Ilustración 62: ssh enviar comando.....	142
Ilustración 63: ssh guardar script.....	143
Ilustración 64: ssh ejecutar guardado.....	144
Ilustración 65: gps inicia desde sms.....	145
Ilustración 66: gps guardar localización.....	146
Ilustración 67: gps enviar correo electrónico.....	147
Ilustración 68: gps enviar sms.....	148
Ilustración 69: gps autoarranque.....	149
Ilustración 70: control monitorizar.....	150
Ilustración 71: control descargar.....	151
Ilustración 72: control filtrar.....	153
Ilustración 73: control quitar filtro.....	154
Ilustración 74: control borrar.....	155
Ilustración 75: Borrar notificación.....	156
Ilustración 76: Hay notificaciones.....	156
Ilustración 77: Obtener notificaciones.....	156
Ilustración 78: Scan Log.....	157
Ilustración 79: Diagrama componentes.....	164
Ilustración 80: Diagrama componentes completo.....	165
Ilustración 81: Página de descarga de Java JDK.....	173
Ilustración 82: Página de descarga de Android SDK.....	174
Ilustración 83: Instalación de plataformas de Android.....	175
Ilustración 84: Página de descarga de Eclipse.....	175
Ilustración 85: Instalación Plugin ADT.....	176
Ilustración 86: Instalación librerías externas.....	177
Ilustración 87: Página principal de Apache Tomcat 7.0.....	179
Ilustración 88: Página principal de Apache Axis2.....	180
Ilustración 89: Iniciar ADV Manager.....	182
Ilustración 90: Configuración dispositivo virtual Android.....	183
Ilustración 91: Compilación aplicación Android.....	183

---

Ilustración 92: Menú desplegable para creación de contenedor de Web Services.....	184
Ilustración 93: Selección de Web Service.....	185
Ilustración 94: Elección Web Service Runtime.....	186
Ilustración 95: Configuración Web Service.....	187
Ilustración 96: Iniciar servidor.....	188
Ilustración 97: Menú desplegable compilación Web Service.....	189
Ilustración 98: Selección de servidor donde correr el Web Service.....	190
Ilustración 99: Añadir la clase del Web Service.....	191
Ilustración 100: Pantalla principal Apache Axis2.....	192
Ilustración 101: Listado Web Services publicados.....	192
Ilustración 102: Creación archivo aar para Web Service.....	193
Ilustración 103: Menú desplegable para exportación.....	194
Ilustración 104: Selección de exportación a archivo JAR.....	195
Ilustración 105: Datos de exportación del demonio.....	196
Ilustración 106: Creación de grupo de hilos.....	199
Ilustración 107: Configuración del numero de hilos.....	200
Ilustración 108: Añadir muestreador Web Service.....	200
Ilustración 109: Configuración datos Web Service.....	201
Ilustración 110: Añadir Gráfico de Resultados.....	201
Ilustración 111: Resultado de las pruebas de stress.....	202
Ilustración 112: Web Service publicado.....	211
Ilustración 113: Pantalla Login.....	215
Ilustración 114: Pantalla falsa.....	216
Ilustración 115: Pantalla Principal.....	217
Ilustración 116: ssh manual.....	218
Ilustración 117: ssh favoritos.....	218
Ilustración 118: Consola.....	219
Ilustración 119: Últimos ejecutados.....	220
Ilustración 120: Añadir último.....	221
Ilustración 121: Ejecutar favorito.....	222
Ilustración 122: Preferencias GPS.....	224
Ilustración 123: GPS principal.....	225
Ilustración 124: Control ajustes.....	226
Ilustración 125: Formulario control.....	227
Ilustración 126: Lista servidores.....	228
Ilustración 127: Actualizando notificaciones.....	229
Ilustración 128: Filtro.....	230
Ilustración 129: Borrando notificaciones.....	231

## Índice de tablas

Tabla 1: Comparación aplicaciones.....	34
Tabla 2: Costes de personal.....	48
Tabla 3: Costes de software.....	50
Tabla 4: Costes generales.....	51
Tabla 5: Coste total.....	51
Tabla 6: Plantilla de requisitos.....	59
Tabla 7: RF01 – Obtención de archivos log.....	59
Tabla 8: RF02 – Análisis de los archivos log.....	60
Tabla 9: RF03 – Escucha de peticiones de información.....	60
Tabla 10: RF04 – Control de notificaciones descargadas en dispositivo.....	60
Tabla 11: RF05 – Borrar notificaciones para dispositivos.....	60
Tabla 12: RF06 – Notificaciones de acceso.....	61
Tabla 13: RF07 – Comunicación SSH.....	61
Tabla 14: RF08 – Servidores recientes.....	61
Tabla 15: RF09– Almacenamiento de comandos.....	61
Tabla 16: RF10– Creación de scripts.....	62
Tabla 17: RF11 – Envío de coordenadas.....	62
Tabla 18: RF12 – Almacenamiento de localizaciones.....	62
Tabla 19: RF13 – Ejecución en background.....	62
Tabla 20: RF14 – Activación remota.....	63
Tabla 21: RF15 – Borrado de datos.....	63
Tabla 22: RF16 – Envío instantáneo.....	63
Tabla 23: Plantilla de Casos de Uso.....	69
Tabla 24: Caso de Uso Consola Android SSH.....	70
Tabla 25: Caso de Uso Localización GPS.....	72
Tabla 26: Caso de Uso Control de Accesos a Servicios.....	74
Tabla 27: Caso de Uso Gestión de Filtros.....	76
Tabla 28: Caso de Uso Gestión de Filtros.....	77
Tabla 29: Pruebas Control de Accesos.....	81
Tabla 30: Pruebas Terminal SSH.....	81
Tabla 31: Pruebas Localizador GPS.....	82
Tabla 32: Notificación.....	96
Tabla 33: Dispositivo.....	96
Tabla 34: Notificación-Dispositivo.....	96
Tabla 35: Servidores.....	97
Tabla 36: Scripts.....	97
Tabla 37: Comandos.....	97
Tabla 38: Coordenada.....	98
Tabla 39: Servidores.....	98
Tabla 40: Email.....	98
Tabla 41: Referencia a los requisitos.....	159
Tabla 42: Pruebas principal.....	160
Tabla 43: Pruebas ssh accesodatos.....	161

---

Tabla 44: Pruebas ssh vista.....	162
Tabla 45: Pruebas gps vista.....	163
Tabla 46: Pruebas accesodatos vista.....	164
Tabla 47: Pruebas webservice.....	164
Tabla 48: Pruebas sobre Launcher.....	201
Tabla 49: Pruebas sobre Terminal SSH.....	201
Tabla 50: Pruebas sobre Control de accesos.....	202
Tabla 51: Pruebas sobre Localizador GPS.....	203



# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

**Parte I: Descripción del proyecto**

**Alumnos** David Herrero de la Peña  
Jonatan Santos Barrios

**Tutor** Carlos López Nozal  
DEPARTAMENTO DE INGENIERÍA CIVIL  
Área de Lenguajes y Sistemas Informáticos  
Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## Índice de contenido

1. Introducción.....	20
1.1. Estructura de la memoria.....	20
2. Objetivos del proyecto.....	21
2.1. Objetivos principales.....	21
2.2. Objetivos técnicos.....	21
3. Conceptos teóricos.....	22
3.1. Gestión de servidores.....	22
3.2. Web Service.....	22
3.3. GPS.....	23
3.4. Syslog.....	24
3.5. Estilos de diseño de aplicaciones Android.....	24
3.6. Scrum.....	25
3.7. Sistema de control de versiones.....	26
3.8. Sistema de control de tareas.....	26
4. Técnicas y herramientas.....	27
4.1. Lenguaje Java.....	27
4.2. Android SDK.....	28
4.3. SQLite.....	28
4.4. MySQL.....	29
4.5. Apache Axis2.....	29
4.6. Apache Tomcat.....	29
4.7. JSVC.....	30
4.8. JSCH.....	30
4.9. Eclipse.....	30
4.10. Git.....	31
4.11. GitHub.....	31
4.12. TortoiseGIT.....	31
4.13. Open Office.....	32
4.14. Astah Community.....	32
4.15. JMeter.....	32
4.16. Droid Explorer.....	32
5. Aspectos relevantes del desarrollo.....	32
5.1. Versiones de Android.....	32
6. Trabajos relacionados.....	33
7. Conclusiones y líneas de trabajo futuras.....	34
7.1. Conclusiones.....	34
7.2. Líneas de trabajo futuras.....	35

## **Índice de ilustraciones**

Ilustración 1: Comunicación Web Service [5].....	23
Ilustración 2: Diagrama de la metodología Scrum [13].....	26
Ilustración 3: Distribución de las versiones de Android en Junio de 2012 [30].....	33

## **Índice de tablas**

Tabla 1: Comparación aplicaciones.....	34
--	----

## **LISTA DE CAMBIOS**

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>
1.0	21/02/12	Primera Entrega: Introducción, Objetivos, Conceptos Teóricos y Técnicas y Herramientas.
2.0	02/03/12	Segunda Entrega: Revisión y Ampliación, Aspectos Relevantes y Trabajos Relacionados.
3.0	19/03/12	Tercera Entrega: Revisión y Ampliación.
4.0	20/06/12	Cuarta Entrega: Finalización del documento. Conclusiones y líneas de trabajo futuras.

## 1 INTRODUCCIÓN

El objetivo de este proyecto es el desarrollo de una aplicación que permita la gestión y administración remota de varios servidores de una misma organización, la aplicación irá dividida en tres partes bien diferenciadas, una de ellas permitirá llevar un control a distancia de los accesos de los usuarios que se autentificado en los servidores de una organización todo ello en tiempo real, así como también un control del servidor de correo, con avisos de spam o intento de robo de cuenta de correo, otra de las partes de la aplicación es la gestión a distancia de los mismo servidores, pudiendo realizar cualquier operación de administración desde el terminal, y por otra parte la localización y seguimiento de la posición de los terminales Android de los que dispone la organización.

La deslocalización de las empresas hace imposible la gestión de todos los ordenadores de una empresa, situados en distintos puntos de la geografía mundial, por lo que se crea la necesidad de una herramienta que permita gestionar a distancia todos ellos desde un solo terminal.

El control de accesos a los servidores es una de las partes más importantes en temas de seguridad, por ello se hace necesaria una herramienta que informe de los accesos e intentos de ataque en tiempo real de los distintos servidores de una misma organización.

A la hora de enfrentarse al problema de la perdida de un dispositivo que contenga información importante y confidencial para la organización, disponemos de la ayuda de los sistemas de posicionamiento GPS, para poder localizar un dispositivo móvil, estos sistemas necesitan de un receptor GPS, con el que cuenta en la actualidad la totalidad de los dispositivos Android, entonces se podrá almacenar la información recogida por los terminales y luego ser enviada donde se necesite, para procesarla y determinar donde se encuentra el terminal.

### 1.1 Estructura de la memoria

Esta memoria se ha desarrollado siguiendo la estructura descrita en [Peñalvo et al., 1999].

Introducción: Descripción breve tanto del tema del proyecto como del contenido y la estructura de la memoria.

Objetivos del proyecto: Explicación concisa de cuáles son los objetivos que se persiguen con la realización del proyecto.

Conceptos teóricos: Explicación de conceptos específicos que sean necesarios a la hora de desarrollar y comprender el proyecto.

Técnicas y herramientas: Aspectos más destacados de las técnicas metodológicas y las herramientas de desarrollo utilizadas para la realización del proyecto.

Aspectos relevantes del desarrollo del proyecto: Descripción de aspectos destacables sobre las fases de análisis, diseño e implementación.

Trabajos relacionados: Estado del arte; descripción de trabajos y proyectos realizados

con anterioridad en el campo del proyecto que se está realizando.

Conclusiones y líneas de trabajo futuras: Resumen de las conclusiones obtenidas con la realización del proyecto e informe de las posibles formas de mejorar el proyecto o seguir trabajando en la línea del proyecto realizado.

Plan del proyecto software: Planificación temporal y estudio de viabilidad del proyecto.

Especificación de requisitos del software: Este apartado recoge la fase de análisis; la descripción del sistema actual, el catálogo de requisitos del sistema y la especificación de requisitos funcionales, de información, de eventos y no funcionales.

Especificación de diseño: Se describe la fase de diseño; el ámbito del software, el diseño de datos, el diseño arquitectónico, el diseño de la interfaz, el diseño procedural, el entorno tecnológico del sistema y el plan de desarrollo e implementación.

Documentación técnica de programación: Parte correspondiente al código de la aplicación; contiene la documentación de las bibliotecas, el código fuente, el manual del programador y las pruebas unitarias.

Manuales de usuario: Recoge todos los manuales, tutoriales y guías de usuario necesarios para el manejo de la aplicación. Son importantes el documento de instalación y configuración y el manual de usuario.

## **2      OBJETIVOS DEL PROYECTO**

### **2.1    Objetivos principales**

Los objetivos principales del proyecto son tres:

- Diseñar un sistema que permita la gestión de los servidores de una organización, de forma remota mediante un dispositivo con Android.
- Diseñar un sistema que notifique a un terminal Android los accesos e intentos de ataque a los servidores de la organización.
- Diseñar un sistema que permita la localización y seguimiento de un dispositivo móvil Android.

### **2.2    Objetivos técnicos**

Los objetivos técnicos son:

- El proyecto debe trabajar en sistemas operativos Android.
- El lenguaje de programación para el desarrollo del proyecto es Java.
- Para generar la documentación del código se utilizará la herramienta JavaDoc.

### 3 CONCEPTOS TEÓRICOS

#### 3.1 Gestión de servidores

La gestión adecuada de los servidores es un tema que cobra vital importancia en las organizaciones que cuentan con tecnologías de la información. Los administradores se enfrentan a la tarea de administrar eficientemente la infraestructura que cubre las necesidades de la empresa, las organizaciones necesitan que dicha infraestructura este disponible y operativa las 24 horas del día, durante todo el año, lo que requiere plena dedicación por parte de los administradores de dicha infraestructura, para vigilar el correcto funcionamiento.

Dentro de una empresa puede existir multitud de equipos servidores, y ademas cabe la posibilidad que dichos servidores se encuentre físicamente situados a cientos de kilómetros. Lo que hace muy complicado la gestión y administración de los servidores.

Una solución parcial para el problema de la distancia, es la administración remota por medio de programas de control remoto, pero esta solución requiere de la disponibilidad de un ordenador cerca para realizar las operaciones pertinentes.

Por lo cual, existe la necesidad de disponer de un método que no requiera la necesidad de un ordenador, es donde aparece la figura de los dispositivos móviles, con muchas más opciones de portabilidad y con recursos suficientes para poder realizar las tareas de administración y gestión de los servidores remotamente.

#### 3.2 Web Service

Existen múltiples definiciones de Web Service, por ejemplo la organización World Wide Web Consortium (W3C) [1] [2], que establece las normas para los servicios Web, los define así: “Un servicio web es un sistema de software identificado por un URI, cuya interfaz pública y enlaces se definen y describen mediante XML. Su definición puede ser descubierta por otros sistemas de software. Estos sistemas pueden interactuar con el servicio Web utilizando mensajes basados en XML transportados por protocolos de Internet.” [3].

Otra definición más sencilla y quizás más útil es: “Un Web Service es un método de comunicación entre dos aplicaciones a través de Internet, aún en el caso de que estas estén programadas sobre tecnologías y lenguajes diferentes.” La interfaz de los Web Services esta definida por mensajes que acepta y devuelve, por lo que los consumidores solo tiene que poder crear y consumir dichos mensajes, lo que permite implementarlo en cualquier plataforma y lenguaje de programación.

El término Web Services describe una forma estandarizada de integrar aplicaciones web mediante el uso de XML, SOAP, WSDL y UDDI, sobre los protocolos de Internet. XML es usado para describir los datos, SOAP se ocupa de la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa de conocer cuales son los servicios disponibles [4].

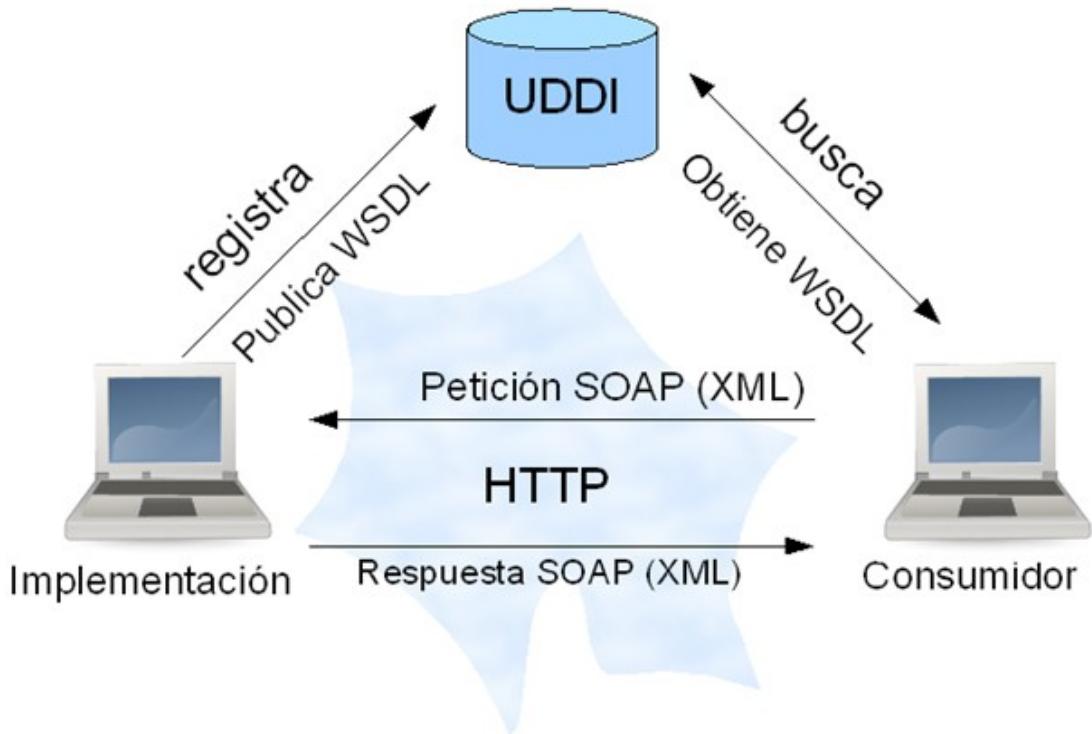


Ilustración 1: Comunicación Web Service [5]

La comunicación en un servicio web puede ir desde un simple paso de mensajes entre los servicios a un escenario complejo en el que un conjunto de servicios en conjunto se coordinen entre sí para lograr un objetivo común.

Una de las razones de uso de Web Services es que se pueden utilizar con HTTP sobre TCP a través del puerto 80, se utiliza dicho puerto porque las organizaciones por razones de seguridad utilizan firewalls, y bloquean la mayoría de los puertos TCP, salvo el puerto 80, que es usado por los navegadores webs.

A la hora del desarrollo del proyecto se plantea la necesidad del uso de este tipo de tecnología, puesto que es necesario que dos aplicaciones se comuniquen entre ellas, a través de la web, y aplicando un cierto nivel de seguridad.

### 3.3 GPS

El GPS (*Global Positioning System*) es un sistema de navegación por satélite que permite determinar la posición en la tierra de un objeto con un margen de error de unos pocos metros. Algunas de las ventajas del uso de esta tecnología son que funciona en cualquier condición climática, en cualquier parte del mundo, las 24 horas del día y ademas su uso es gratuito [6].

El sistema esta compuesto por 24 satélites de radionavegación repartidos alrededor de la tierra a unos 20.000 kilómetros de altura. Dichos satélites emiten informaciones permanentes que permiten a cualquier receptor GPS determinar su posición [7].

Sin embargo el GPS presenta dificultades a la hora de calcular la posición en entornos con baja señal, como pueden ser en ciudades con grandes edificios, que dificulten la recepción de las señales, para solucionar este problema aparecieron los A-GPS o GPS asistidos, los que se encuentran en la mayoría de dispositivos móviles, este tipo de receptores se ayudan de datos proporcionados por un Servidor de Asistencia, por lo que tienen la necesidad de disponer de conexión de datos activa con el consiguiente coste de conexión.

La tecnología A-GPS esta pensada principalmente para acelerar el proceso de inicio del dispositivo, un proceso costoso en tiempo, ya que requiere de unos complejos cálculos en localizar la posición inicial del dispositivo. El A-GPS no es una tecnología distinta, sino un añadido al GPS que reduce el tiempo de iniciación.

En la actualidad, prácticamente la totalidad de los dispositivos móviles Android disponen de GPS incorporado, y en lo que respecta a este proyecto se necesitará hacer uso del GPS que venga incorporado con el dispositivo, para obtener el posicionamiento de los diferentes dispositivos móviles Android de los que dispone la organización, para llevar un control exhaustivo de toda la flota de dispositivos, y en caso de perdida o extravío, poder localizarlo.

### 3.4 Syslog

Syslog es un estándar de IETF (Internet Engineering Task Force) para el envío de mensajes de registro de un red informática. Por syslog se conoce tanto al protocolo de red como a la aplicación encargada de enviar los mensajes. Puede ser utilizado para diferentes tareas: gestión del sistema informático, auditoria de seguridad, o análisis de la información y mensajes de depuración.

El protocolo utilizado por syslog se define en la RFC5424 [8].

La utilización de una aplicación o librería que realice la tarea de recepción y registro de todos los mensajes es imprescindible para poder llevar el control de acceso a los equipos en el actual proyecto.

### 3.5 Estilos de diseño de aplicaciones Android

Dado que Android es un proyecto de código abierto, desde su nacimiento ha tenido el problema de una cierta falta de estilo en lo que se refiere a la apariencia de sus aplicaciones, ya que cada desarrollador aplicaba el estilo que le parecía más adecuado a sus aplicaciones; es por ello que a principios de enero de 2012, la Open Handset Alliance [9], los padres de Android, liderados por Google, sacan una guía con las directrices y pautas para la creación de interfaces de usuario [10], todo ello para hacer que las aplicaciones se comporten de una manera más predecible y consistente, y que estén en armonía con el resto del sistema Android.

Esta guía ofrece a los desarrolladores:

- Una visión general de los estilos de Android.
- Los patrones de diseño que se pueden aplicar.
- Como establecer los elementos visuales.

- El momento en el que mostrar las notificaciones.
- Como interpretar los gestos del usuario sobre la pantalla.
- La adaptación de una versión de Android a otra.

La aplicación de esta guía cobra vital importancia en este proyecto, pues al hacer la aplicación más familiar e intuitiva al usuario, y con un funcionamiento fluido, se hace más agradable su uso y mayor probabilidad de éxito en el mercado, ya que los usuarios suelen ser reticentes a los cambios.

### 3.6 Scrum

Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprints. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija, terminan en una fecha específica aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. Todos los días el equipo se reúne brevemente para informar del progreso, y actualizan unas gráficas sencillas que les orientan sobre el trabajo restante. Al final del Sprint, el equipo revisa el Sprint con los interesados en el proyecto, y les enseña lo que han construido. La gente obtiene comentarios y observaciones que se puede incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar [11].

Los roles principales en Scrum son el ScrumMaster que mantiene los procesos y trabaja junto con el jefe de proyecto, el Product Owner que representa a las personas implicadas en el negocio y el Team que incluye a los desarrolladores.

Scrum permite la creación de equipos con organización propia fomentando la localización conjunta de todos los miembros del equipo y la comunicación verbal entre todos los miembros del equipo y las disciplinas implicadas en el proyecto.

Un principio la clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar sus pensamientos sobre lo que quieren y necesitan [12].

Una parte muy importante de Scrum son las reuniones que se realizan durante cada una de las iteraciones. Hay distintos tipos:

- Scrum diario: cada día durante la iteración, tiene lugar una reunión de estado del proyecto. A esta reunión se le domina Scrum.
- Reunión de planificación de iteración (Sprint): se lleva a cabo al principio del ciclo de la iteración.
- Reunión de revisión de iteración: al final del ciclo de la iteración.

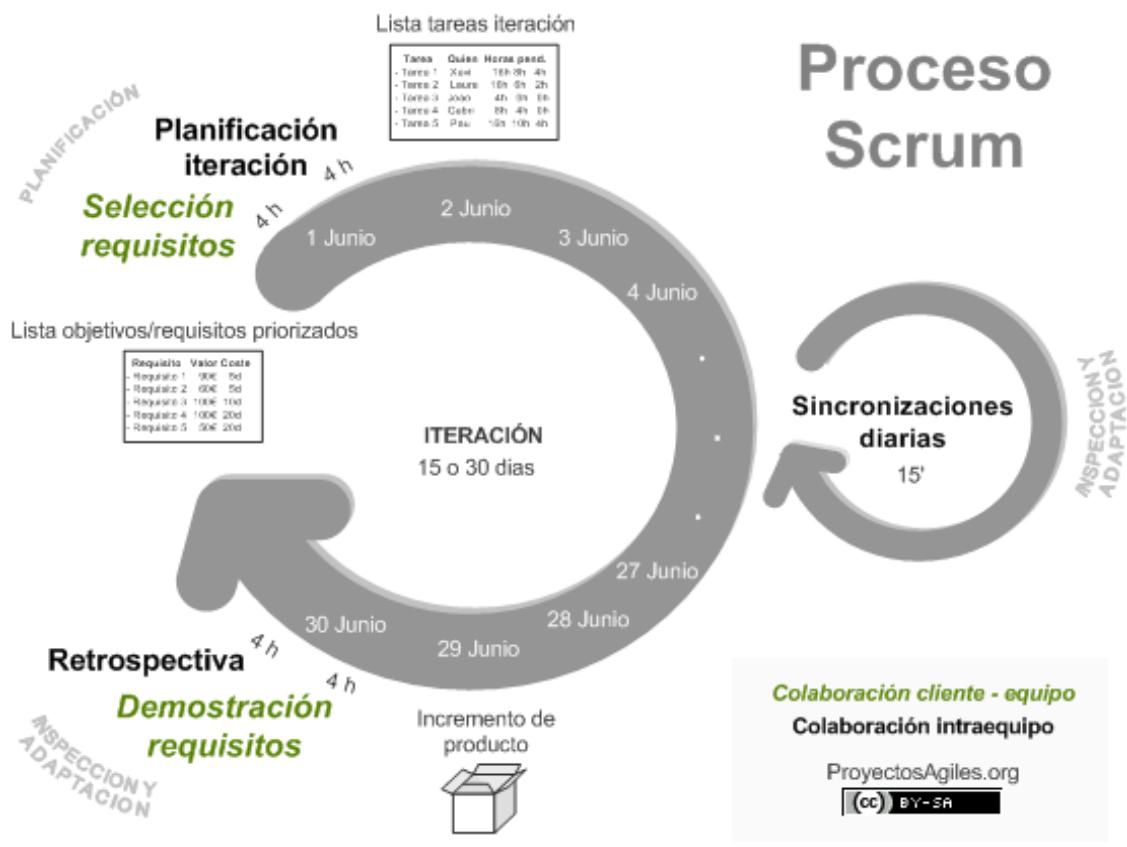


Ilustración 2: Diagrama de la metodología Scrum [13]

### 3.7 Sistema de control de versiones

Los sistemas de control de versiones son sistemas que sirven para mantener distintas versiones de un fichero, normalmente código fuente, documentación o ficheros de configuración.

El sistema registra los cambios realizados sobre los ficheros a lo largo del tiempo, creando diferentes versiones, de manera que se pueda recuperar cualquier versión anterior.

Estos sistemas utilizan una arquitectura cliente–servidor, en el servidor se guardan todas las versiones del proyecto. Nos permiten obtener un historial de cambios del proyecto, unir cambios de distintos usuarios sobre el mismo fichero, y gestionar conflictos producidos al modificar simultáneamente dos o más usuarios la misma porción de código de un fichero.

### 3.8 Sistema de control de tareas

Los sistemas de control de tareas se tratan de sistemas que nos ayudan a asegurar que las tareas realizadas se ajustan a las tareas que se fueron planificadas en su momento, y permite llevar un control de progreso de los proyectos.

Un sistema de control de tareas ofrece:

- Una visión general de todas las tareas del proyecto.

- Conocimiento de los tiempo planificados y empleados en cada tarea.
- Posibilidad de calcular los costes, a partir de los tiempos empleados en cada tarea.

## 4 TÉCNICAS Y HERRAMIENTAS

Ahora se detallará las diferentes herramientas utilizadas para la realización del proyecto, todas ellas son software libre y pueden ser utilizadas por cualquier persona.

### 4.1 Lenguaje Java

Java [14] es un lenguaje de programación orientado a objetos, fue desarrollado por la empresa Sun Microsystem a principio de los años 90. Sun Microsystem en 2007 liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL. Java toma mucha de su sintaxis de C y C++, por lo que un programador que haya aprendido cualquiera de estos dos lenguajes encontrará relativamente sencillo aprender a programar en Java. Aunque tome mucha de la sintaxis de C y C++ elimina alguna de sus características de más bajo nivel, como son los punteros y la manipulación directa de la memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Dos de las características más importantes que se buscaba al desarrollar Java eran:

- Debería usar el paradigma de programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.

La primera característica, orientado a objetos, se refiere a un método de programación y al diseño del lenguaje. La idea principal de la programación orientada a objetos es la de unir en un mismo sitio las operaciones y sus datos, esto es lo que forma la entidad llamada objeto. Esta separación en objetos de los datos y las operaciones que están relacionadas entre sí hace más sencilla la tarea de diseñar un sistema software, ya que se asemeja más a la vida real.

Estos objetos, compuestos de datos y operaciones, son relativamente independientes entre ellos, al final, el software completo está compuesto por un conjunto de objetos que se envían mensajes unos a otros, se crean y se destruyen. Otras ventajas de este paradigma de programación derivan de esta última característica, al estar dividido el sistema en diferentes clases, dividir el trabajo entre los diferentes miembros del equipo es mucho más sencillo, además, se fomenta en gran medida la reutilización del código, ya que un mismo objeto se puede utilizar en diferentes ocasiones. Esta última ventaja se apoya además en otras características de Java, como son la herencia y la genericidad. Gracias a estas características es más sencillo crear objetos genéricos capaces de ser utilizados por más y más sistemas software distintos.

La segunda gran característica del lenguaje Java, la independencia de la plataforma, se consigue, a grandes rasgos, mediante la utilización de una máquina virtual (JVM) escrita en código nativo de la plataforma destino (existe una JVM diferente para cada plataforma, por ejemplo, la JVM de Linux y de Windows son diferentes). Al compilar el

código fuente escrito en Java se genera un código conocido como “bytecode”, el cual está a medio camino entre el código fuente y el código máquina. El bytecode es ejecutado por la JVM, que interpreta y ejecuta el código. Además se suministran bibliotecas adicionales (API’s) para acceder a las características de cada dispositivo (como los gráficos, interfaz de red) de forma unificada.

Utilizamos lenguaje Java debido a que es la forma nativa de programar en dispositivos móviles con sistema operativo Android. De esta forma la ejecución de las aplicaciones es mucho mas fluida y su el propio S.O puede manejar mejor la ejecución de las mismas. Además es mucha la documentación que se proporciona desde la propia empresa Google lo que hace que se encuentre toda la información que se necesita.

## 4.2 Android SDK

Android SDK [15] son las siglas de “Android Software Development Kit”. Es un kit de desarrollo que proporciona Google con el que podremos desarrollar aplicaciones para su sistema operativo Android. El SDK es imprescindible para utilizar el controlador ADB (Android Debugging Bridge), herramienta mediante la cual podremos controlar un teléfono Android desde el Pc. Android SDK es distribuido bajo la licencia Apache Software License, 2.0.

Sin embargo lo que a nosotros como desarrolladores de aplicaciones Android nos interesa más es el conjunto de librerías que se incluyen en el SDK para trabajar con el hardware de un teléfono móvil, además del AVD “Android Virtual Device”, utilidad que nos permite crear dispositivos Android virtuales para instalar nuestras aplicaciones y probarlas sin necesidad de disponer de un teléfono Android o una tableta Android.

En la realización de este proyecto hemos tenido que utilizar los dispositivos virtuales que podemos crear con este SDK ya que resulta mucho mas cómodo ejecutar la aplicación en un dispositivo virtual que tener que instalar la aplicación en un dispositivo real para probarla. Además hemos utilizado varias librerías para realizar las conexiones ssh (Secure SHell) y la utilización del GPS.

## 4.3 SQLite

SQLite [16] es un sistema de gestión de bases de datos relacional contenida en una relativamente pequeña biblioteca escrita en C (unos 275 Kbs y unas 25.000 líneas de código), SQLite es Software Libre y todo el código fuente es de dominio público.

Una de las principales diferencias de SQLite respecto a otros sistemas gestores de datos cliente-servidor como por ejemplo Oracle, es que SQLite no arranca un servicio independiente con el que nuestro programa debería comunicarse para tener acceso a la base de datos, si no que la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. Nuestro programa entonces utiliza llamadas a funciones y rutinas para manejar la base de datos.

SQLite es muy rápido y ocupa muy poco espacio, además de que no consume recursos ya que no inicia ningún servicio propio como hemos dicho antes, esto lo hace perfecto para utilizarlo en dispositivos móviles donde se premia la velocidad y el menor consumo. SQLite no permite conexiones simultáneas de escritura en la misma base de

datos, pero eso en una aplicación móvil no importa en absoluto ya que solo se estará ejecutando una tarea a la vez y nadie mas que nuestro propio programa necesitará acceder a esa base de datos.

Otra de las razones por las que hemos utilizado SQLite es que está perfectamente integrado con dispositivos móviles Android gracias a las librerías que se suministran con el SDK.

#### 4.4 MySQL

MySQL [17] es un sistema de gestión de base de datos relacional, multihilo y multiusuario, es propiedad de Sun Microsystems esta a su vez propiedad de Oracle, se trata de un software de distribución gratuita bajo la licencia GNU General Public License v2.

MySQL es un gestor de base de datos muy rápido en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Se utilizó MySQL como gestor de base de datos, puesto que se adecua a nuestras necesidades para dar servicio de almacenamiento a la aplicación web, por su rapidez y que no utiliza los mínimos recursos del equipo que hace de servidor.

#### 4.5 Apache Axis2

Apache Axis2 [18] es un contenedor de Web Services, es desarrollado por Apache Software Foundation y distribuido con licencia Apache License 2.0.

Apache Axis2 proporciona un modelo de objetos y arquitectura modular que facilita la tarea de añadir funcionalidad y dar soporte para nuevos servicios Web. Axis2 dispone de una estructura modular que permite ampliar la funcionalidad básica del sistema.

Se hace necesaria la utilización de este software en el proyecto, para la publicación del servicio web necesario en los servidores para intercambiar información con las aplicación principal instalada en el dispositivo móvil.

#### 4.6 Apache Tomcat

Apache Tomcat [19] es un servidor de aplicaciones Web o servlets desarrollado bajo el proyecto Jakarta por Apache Software Foundation, distribuido bajo la licencia Apache License 2.0.

Tomcat es un servidor web con soporte de servlets y JSP (JavaServer Pages). Ya que Tomcat es desarrollado en Java, puede funcionar en cualquier sistema operativo que disponga de la máquina virtual Java.

Con el contenedor de servicios web Apache Axis2, no es suficiente para poder hacer uso del servicio web, para poder publicar servicios web se usa una combinación Apache Tomcat + Apache Axis2.

## 4.7 JSVC

Jsvc o Common Daemon [20] es un conjunto de bibliotecas y aplicaciones para hacer que las aplicaciones Java se ejecuten en UNIX con más facilidad, Jsvc pertenece al proyecto Apache Commons Project, que es desarrollado por Apache Software Foundation, y distribuido bajo la licencia Apache License 2.0.

Jsvc permite la ejecución a aplicaciones Java realizar algunas operaciones con privilegios como root, su principal función es la de ejecutar aplicaciones Java como Demonios o Servicios de un sistema operativo.

Jsvc consta de dos partes: una biblioteca nativa escrita en C que interactúa con el sistema operativo, y la biblioteca que proporciona la API, escrita en Java, existe dos formas de utilizar Jsvc, implementando la interfaz del demonio o llamando a un clase que proporciona los métodos necesarios para el demonio.

Esta librería es usada en el proyecto para la creación de un demonio que se encargara del control de accesos al servidor, por lo que es necesario que se ejecute como demonio pues es necesario que se este ejecutando continuamente.

## 4.8 JSCH

JSCH [21] es una librería de utilidades de comunicación para Java creada por Jcraft con licencia BSD (licencia de software libre permisiva, se puede usar en el código fuente de software no libre). Esta librería se utiliza principalmente para conexiones SSH2 o SSH1.

Se ha utilizado esta librería para las conexiones ssh ya que es muy sencilla de utilizar en comparación con otras librerías de conexión ssh, además de la poca dependencia con otras librerías para su funcionamiento. Al ser una aplicación en un dispositivo móvil, el hecho de que necesite menos librerías adicionales repercute en el tamaño final de la aplicación, el cual siempre intentamos minimizar.

## 4.9 Eclipse

Eclipse [22] es un IDE (entorno de desarrollo integrado) de código abierto y multiplataforma. Eclipse fue desarrollado originalmente por IBM, aunque actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto.

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

Como cualquier IDE, eclipse cuenta con un editor de textos (que además permite el uso de auto-completado), un compilador/interprete y un depurador de código.

En principio se diseñó para programar en lenguaje Java, y según las estadísticas es el lenguaje en el que más líneas de código se han escrito con Eclipse, pero gracias a una de las características más interesante de Eclipse es posible utilizar este IDE para programar en C, PHP, etc. La característica de la que hablamos son los plugins, o extensiones,

gracias a los cuales es posible añadir funcionalidad de forma dinámica a Eclipse.

Para este proyecto se ha usado Eclipse, ya que Google suministra un plugin para Eclipse que permite integración con el SDK de Android, con lo cual todo el desarrollo de la aplicación es mucho más sencillo.

## 4.10 Git

Git [23] es un sistema de control de versiones distribuido diseñado por Linus Torvalds, Git es desarrollado bajo licencia GNU GPL v2. Está diseñado para manejar proyectos muy grandes con velocidad y eficiencia, pero igual de apropiado para repositorios pequeños, como en el caso de nuestro proyecto.

Cada directorio de trabajo de Git es un repositorio completo con historial y capacidades totales de tracking de revisiones, independiente del acceso a la red o de un servidor central.

La razón de elegir Git en vez de otros sistemas de control de versiones como por ejemplo Subversion es el hecho de que Git es mucho mas rápido, al utilizar scrum en el proceso de desarrollo y que el código de la aplicación cambia muchas veces la velocidad se convierte en un punto crítico.

## 4.11 GitHub

GitHub [24] es un servicio de alojamiento de repositorios de software que usa el sistema de control de versiones Git.

Aunque Git es un sistema descentralizado (no necesita de un servidor donde subir los cambios, si no que son los componentes del equipo los que se envían los cambios los unos a los otros), es muy habitual que se trabaje mejor si existe un servidor común a todos donde alojar el repositorio de código en el que trabaja todo el equipo, y esta es la principal tarea de GitHub.

GitHub no solo actúa como servidor central donde alojar el código, si no que con el paso del tiempo se ha convertido en un lugar de encuentro para programadores de todo el mundo donde se resuelven dudas en el campo de programación y donde se puede aprender del trabajo de otros desarrolladores, ya que la mayoría de repositorios que se almacenan en GitHub son públicos.

Una de las razones de usar GitHub, además de su facilidad de uso, es su popularidad, casi todo el mundo trabaja con él, así que este proyecto nos ayudará a prepararnos para el trabajo fuera de la universidad.

## 4.12 TortoiseGIT

TortoiseGit [25] es un cliente para Git, con interfaz gráfica, y que se fusiona con el explorador de archivos de Windows, lo cual hace que trabajar con él sea muy intuitivo.

Es un programa de Software Libre bajo licencia GNU General Public Licence.

### 4.13 Open Office

OpenOffice [26] es una suite ofimática libre (código abierto y distribución gratuita) que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, base de datos, etc. Está disponible para varias plataformas, por ejemplo GNU/Linux, Solaris, Windows, Mac OS. Soporta más de 110 idiomas. Puede trabajar tanto con ficheros creados por la propia suite, como con ficheros creados por otros programas, como por ejemplo la suite ofimática de Microsoft Office. El hecho de que pueda trabajar con casi cualquier tipo de formato, y que sea de distribución gratuita ha posicionado a esta suite como una clara competidora frente a otras suites de pago.

Esta suite permite añadir extensiones, con las cuales es posible añadir más funcionalidad a la suite.

### 4.14 Astah Community

Astah Community [27] que formalmente es conocida como JUDE es una herramienta de modelado UML de la compañía Change Visión, permite realizar diferentes diagramas como son UML, Casos de Uso, Diagramas de Secuencia, Diagramas de colaboración, Diagramas de estado, etc.

### 4.15 JMeter

JMeter [28] es una aplicación de Apache Jakarta, distribuida bajo licencia Apache License 2.0 que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.

### 4.16 Droid Explorer

Droid Explorer [29] es una herramienta para administrar dispositivos Android y es distribuida bajo la licencia Microsoft Public License, se hará uso de esta herramienta para monitorizar el dispositivo móvil en la pantalla del ordenador, y para poder sacar capturas del mismo.

## 5 ASPECTOS RELEVANTES DEL DESARROLLO

### 5.1 Versiones de Android

Desde la salida al mercado de Android hasta nuestros días, ha sufrido múltiples actualizaciones lo que ha generado una serie de versiones del sistema operativo Android, estas actualizaciones arreglaban bugs y añaden nuevas funcionalidades.

Atendiendo a las distintas versiones de Android, y a la distribución de las mismas en todo el mercado móvil con sistema operativo Android, se ha decidido desarrollar la aplicación para versiones de Android 2.3 o superiores, ya que las versión más usadas es Android 2.3.3, y así llegar a un mayor numero de usuarios, y obtener una mayor compatibilidad con el mercado móvil actual.

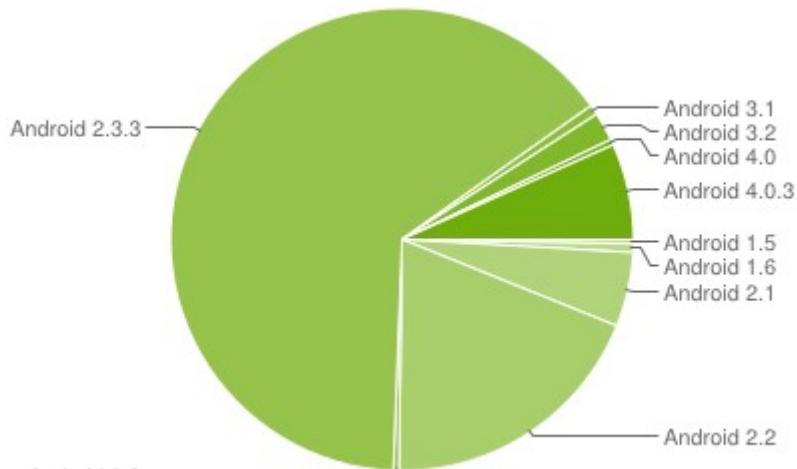


Ilustración 3: Distribución de las versiones de Android en Junio de 2012 [30]

## 6 TRABAJOS RELACIONADOS

En este apartado se muestra algunas aplicaciones que se han encontrado y que realizan funciones parecidas a la aplicación que se quiere desarrollar.

**ConnectBot:** Aplicación que permite conectarse a servidores ssh, telnet o acceder a la consola del equipo donde se ejecute. Permite la creación de túneles seguros y el manejo de puertos (port-forwarding).

**Simple Telnet Client:** Aplicación para acceder a computadores remotos mediante el uso de telnet.

**Cerverus:** Aplicación que permite la localización de un móvil robado. Permite grabación de sonido por el micrófono, tomar fotos con la cámara, posición gps, etc.

**Lookout:** Suite de seguridad que incluye antivirus, backup de los datos y una función para encontrar el teléfono en caso de pérdida.

**SeekDroid:** Aplicación para localizar y controlar el teléfono remotamente. Permite localizar el dispositivo, mostrar la ubicación en un mapa, etc. Todo desde una página web.

**Server Monitor:** Simple monitor para servidores web.

**HTTP Server Monitor:** Monitoriza el estado de los servidores HTTP que se desee. Puede lanzar notificaciones y sms cuando se cumplan algunas condiciones de fallo, como por ejemplo que no responda a un ping en los últimos 5 minutos.

**Bmonitor:** Permite la monitorización de servidores mediante el estado de ping, puertos y http(s).

Todas estas aplicaciones se pueden encontrar en el GooglePlay.

En la siguiente tabla se realiza una comparativa entre todas las aplicaciones que se ha encontrado y nuestra aplicación.

	SSH			GPS			Control			Coste
	Permite ssh	Consola completa	Guarda comandos fav	Control desde exterior	Funcion a sin internet	Transparente al usuario	C. servidor encendido	C. acceso al servidor	C. emails enviados	
ConnectBot	X	X								0€
Simple Telnet Client										0€
Cerverus				X(sms y web)	X	X				2.99€
Lookout				X(web)		X				0€
SeekDroid				X(web y sms)		X				2,24€
Server Monitor							X			0€
HTTP Server Monitor							X			0€
Bmonitor							X			0€
AndroidServerStatus	X		X	X(sms)	X	X	X	X	X	1,99 €

Tabla 1: Comparación aplicaciones

Algunas consideraciones más concretas:

- Tanto Lookout como SeekDroid envían los datos de posicionamiento a un servidor web, por lo que no se puede localizar sin conexión a internet. Nuestra aplicación tiene la opción para enviar las posiciones GPS por sms y por lo tanto no es necesaria una conexión a internet para funcionar.
- Las aplicaciones que realizan el control de servidores son muy sencillas, y no se ha encontrado ninguna que llegue a profundizar tanto en este aspecto como la nuestra. La mayoría solo se dedican a consultar si el servidor está disponible mediante el uso de comandos ping.

## 7 CONCLUSIONES Y LINEAS DE TRABAJO FUTURAS

### 7.1 Conclusiones

El proyecto Android Server Status ha cumplido con todos los requisitos que han sido propuestos a lo largo de su desarrollo. A pesar de que en un principio la aplicación estuviese orientada a otras funcionalidades, el modo de trabajo Scrum nos ha permitido ir adaptando la aplicación a los nuevos requisitos sin demasiado esfuerzo.

La realización de este proyecto nos ha permitido profundizar en casi todos los conocimientos adquiridos durante la carrera, en programación orientada a objetos, patrones de diseño, análisis, bases de datos y administración de sistemas operativos.

Además el hecho de desarrollar la aplicación para una empresa real nos ha permitido comprender como funcionan los proyectos fuera de la carrera, además de que nos ha motivado a seguir mejorando la aplicación en todo lo que podíamos.

El hecho de programar en Android, un sistema operativo del que no habíamos visto nada en la carrera nos ha obligado a buscar mucha información en todas las fuentes posibles y a valorar todo el trabajo que realiza la comunidad de desarrolladores en torno a este sistema operativo.

Este proyecto también nos ha ayudado a trabajar en equipo, ya que el desarrollo se hacía de forma paralela, además de la forma de enfrentarse a los problemas que van surgiendo durante el desarrollo de una aplicación de este tipo y las distintas formas de resolverlos.

Consideramos que el resultado, tanto para nosotros como para los tutores y la empresa que lo pondrá en explotación, ha sido satisfactorio y confiamos en que esta aplicación siga avanzando y mejorándose a lo largo del tiempo.

## 7.2 Líneas de trabajo futuras

La aplicación desarrollada puede ser mejorada y ampliada en varios aspectos, algunos de los que proponemos son:

- **Ampliación de seguridad en el servicio web:** en cuanto a la seguridad del servicio web se han llevado a cabo ciertas medidas de seguridad, como el cambio de puerto por defecto por el que se establece la comunicación, y que no sera un servicio publico y solo tendrá conocimiento de tal servicio el usuario de la aplicación, aun así, proponemos una ampliación para introducir autenticación mediante usuario y contraseña o la implementación de algún tipo de encriptación.
- **Descargar notificaciones en partes:** cuando se descargan las notificaciones en el cliente, es posible que la cantidad de notificaciones sea muy alta y no sea viable estar esperando a que se descarguen todas. Sería posible implementar un sistema de descarga parecido a otras aplicaciones como Twitter en las que se descargan los mensajes según se va avanzando por la lista.
- **Exportar la aplicación a otros S.O:** Sería posible llevar la aplicación a otros sistemas móviles como IOS o RIM.
- **Evitar que la aplicación se puede eliminar:** Evitar que la aplicación de localización del móvil se pueda desinstalar a no ser que se ingrese un código secreto.

## Referencias

- [1] W3C Organization / Tim Berners-Lee – <http://www.w3.org/standards/webofservices/>
- [2] Web Service Activity / W3C Organization - <http://www.w3.org/2002/ws/>
- [3] Designing Web Services with the J2EE 1.4 Platform / Inderjeet Singh, Sean Brydon, Greg Murray, Vijay Ramachandran, Thierry Violeau, Beth Stearns – 2004
- [4] Operating Systems and Middleware: Supporting Controlled Interaction / Max Hailperin Gustavus Adolphus College – 2001
- [5] Web Service / Gramáticas Regulares -  
<http://gramaticasformales.wordpress.com/category/web-services/>
- [6] GPS Fácil, Uso del sistema de posicionamiento global / Lawrence Letham - 2001
- [7] Guía práctica del GPS / Paul Correira – 2000
- [8] The Syslog Protocol / Internet Engineering Task Force - <http://tools.ietf.org/html/rfc5424>
- [9] Open Handset Alliance / Google Inc. - <http://www.openhandsetalliance.com/>
- [10] Android Desing / Google Inc. - <http://developer.android.com/design/index.html>
- [11] Información básica de Scrum / Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde – 2009
- [12] Curso de desarrollo ágil / Laboratorio Nacional de Calidad de Software de INTECO (Instituto Nacional de Tecnologías de la Comunicación) – 2009
- [13] Qué es Scrum / ProyectosAgiles.org - <http://www.proyectosagiles.org/que-es-scrum>
- [14] Java / Oracle Corporation - <http://www.oracle.com/technetwork/java/index.html>
- [15] Android Developers / Google Inc. - <http://developer.android.com/index.html>
- [16] SQLite / D. Richard Hipp - <http://www.sqlite.org/>
- [17] MySQL / Oracle Corporation - <http://www.mysql.com/>
- [18] Apache Axis2 / Apache Software Foundation - <http://axis.apache.org/axis2/java/core/>
- [19] Apache Tomcat / Apache Software Foundation - <http://tomcat.apache.org/>
- [20] JSVC / Apache Software Foundation - <http://commons.apache.org/daemon/jsvc.html>
- [21] JSCH / Jcraft - <http://www.jcraft.com/jsch/>
- [22] Eclipse / The Eclipse Foundation - <http://www.eclipse.org/>
- [23] Git / Linus Torvalds, Juliano Hamano - <http://git-scm.com/>
- [24] GitHub / GitHub, Inc. - <https://github.com/>
- [25] TortoiseGit / Frank Li - <http://code.google.com/p/tortoisegit/>
- [26] OpenOffice / Oracle Corporation - <http://www.openoffice.org/>
- [27] Astah Community / Change Vision - <http://astah.net/editions/community>
- [28] Apache Jmeter / Apache Software Foundation - <http://jmeter.apache.org/>
- [29] Droid Explorer - <http://de.codeplex.com/>
- [30] Platform Versions Android Developers / Google Inc. -  
<http://developer.android.com/resources/dashboard/platform-versions.html>

# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

**Anexo I: Plan del proyecto software**

Alumnos	David Herrero de la Peña Jonatan Santos Barrios
Tutor	Carlos López Nozal DEPARTAMENTO DE INGENIERÍA CIVIL Área de Lenguajes y Sistemas Informáticos Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## **Índice de contenido**

1. Introducción.....	42
2. Planificación temporal.....	42
2.1. Planificación incial.....	42
2.1.1. Fase de análisis.....	42
2.1.2. Fase de diseño.....	43
2.1.3. Fase de implementación.....	43
2.1.4. Fase de pruebas.....	43
2.1.5. Fase de documentación.....	43
2.1.6. Diagrama de Gant de la planificación inicial.....	43
2.1.7. Hoja de tareas de la planificación inicial.....	45
2.2. Planificación final.....	45
2.2.1. Diagrama de Gant de la planificación final.....	46
2.2.2. Hoja de tareas de la planificación final.....	47
3. Estudio de viabilidad.....	47
3.1. Viabilidad económica.....	47
3.1.1. Costes de personal.....	48
3.1.2. Costes de hardware.....	49
3.1.3. Costes de software.....	49
3.1.4. Costes generales.....	51
3.1.5. Costes totales.....	51
3.1.6. Análisis de costes-beneficios.....	52
3.2. Viabilidad legal.....	52

## **Índice de ilustraciones**

Ilustración 4: Diagrama de Gantt de la planificación inicial.....	44
Ilustración 5: Tareas de la planificación inicial.....	45
Ilustración 6: Diagrama de Gantt de la planificación final.....	46
Ilustración 7: Tareas de la planificación final.....	47

## **Índice de tablas**

Tabla 2: Costes de personal.....	48
Tabla 3: Costes de software.....	50
Tabla 4: Costes generales.....	51
Tabla 5: Coste total.....	51

## **LISTA DE CAMBIOS**

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>
1.0	25/03/12	Primera versión: Introducción y planificación temporal.
2.0	12/06/12	Segunda versión: Viabilidad económica.
3.0	18/06/12	Tercera versión: Viabilidad legal.

## 1 INTRODUCCIÓN

En este apartado se recoge toda la información sobre la planificación temporal del proyecto, así como un estudio de la viabilidad del proyecto.

La planificación inicial del proyecto se basa en la supervisión y comparación de los resultados que se van obteniendo con los previstos en la planificación inicial, para comprobar si estamos cumpliendo con los plazos. Y en caso de no cumplir con ello por alguna irregularidad poder realizar las adaptaciones que sean necesarias redistribuyendo u optimizando los recursos. Si no existe esta planificación, es imposible estimar el estado del proyecto en cada momento de forma correcta.

Este apartado también contiene un estudio de viabilidad que aportará un análisis de costes del proyecto, así como la viabilidad técnica, donde se analizará los recursos o requerimientos técnicos que serán necesarios para el proyecto y por ultimo la viabilidad legal del proyecto.

## 2 PLANIFICACIÓN TEMPORAL

Esta fase comienza en el instante de iniciar el proyecto y debería realizarse antes de empezar con otras tareas. El objetivo de la planificación temporal es definir las tareas que compondrán el proyecto y estimar el tiempo que sera necesario invertir para finalizar cada una de ellas.

El marco de desarrollo para la elaboración del proyecto es SCRUM (explicado en el capítulo anterior Descripción del proyecto), un modelo que permite modificar los requisitos previamente establecidos, permitiendo que durante un proyecto los clientes pueden cambiar sus pensamientos sobre lo que quieren y necesitan.

Como herramienta para realizar esta planificación temporal, se empleará el uso de diagramas de Gantt. Esto permite modelar la planificación de las tareas necesarias para la realización del proyecto, y también sirve para crear una representación gráfica del progreso del proyecto.

### 2.1 Planificación incial

El primer objetivo planteado es la finalización del proyecto para el 4 de Junio de 2012. A partir de ahí, se hace una estimación de las iteraciones necesarias a realizar en cada fase y se planifica el tiempo empleado en cada iteración, estableciendo fechas de inicio y fin de cada una.

#### 2.1.1 Fase de análisis

La fase de análisis consiste en establecer los requisitos funcionales de la aplicación y la descripción del sistema. Se establecerán los actores que intervienen en el uso de la aplicación y las acciones más relevantes para el funcionamiento y uso de la aplicación, lo que permitirá realizar una representación gráfica de ellos en diagramas de secuencias.

### **2.1.2 Fase de diseño**

En esta parte se depende de los elementos obtenidos en fase de análisis, y se realiza un modelado de ellos a través de diagramas de diseño arquitectónico, diagramas de clases, o diagramas de secuencias.

Estos diagramas sirven como base del sistema e indican como serán las relaciones entre los diferentes elementos, permitiendo reflejar las funcionalidades de la aplicación que se va a desarrollar de forma gráfica y formal.

### **2.1.3 Fase de implementación**

En esta fase a partir de los modelados obtenidos en la fase de diseño, se empieza a programar e implementar los diseños especificados.

Se programará la aplicación en el lenguaje de programación definido en la fase de análisis, y se estudiará la forma de utilizar las diferentes herramientas que se usarán para el desarrollo.

Se instalarán el entorno de desarrollo del proyecto y todas las herramientas que sean necesarias.

### **2.1.4 Fase de pruebas**

Una vez finalizada la programación de la aplicación se procede a realizar todas las pruebas que sean necesarias para verificar el correcto funcionamiento de la aplicación.

Estas pruebas permiten detectar y corregir errores cometidos a la hora de desarrollar la aplicación.

### **2.1.5 Fase de documentación**

La documentación se generará a medida que va avanzando el proyecto, para que en ella quede reflejada los cambios y tiempos que se han ido produciendo.

Tras acabar con la programación de la aplicación se redactará un manual del programador, que sirva de ayuda a un programador para que pueda continuar con el desarrollo de la aplicación, también se documentará un manual de instalación, indicando los pasos a seguir para la correcta instalación de la aplicación, y por último se redactará un manual de usuario, que sirva como guía para el correcto uso y manejo de la aplicación.

### **2.1.6 Diagrama de Gantt de la planificación inicial**

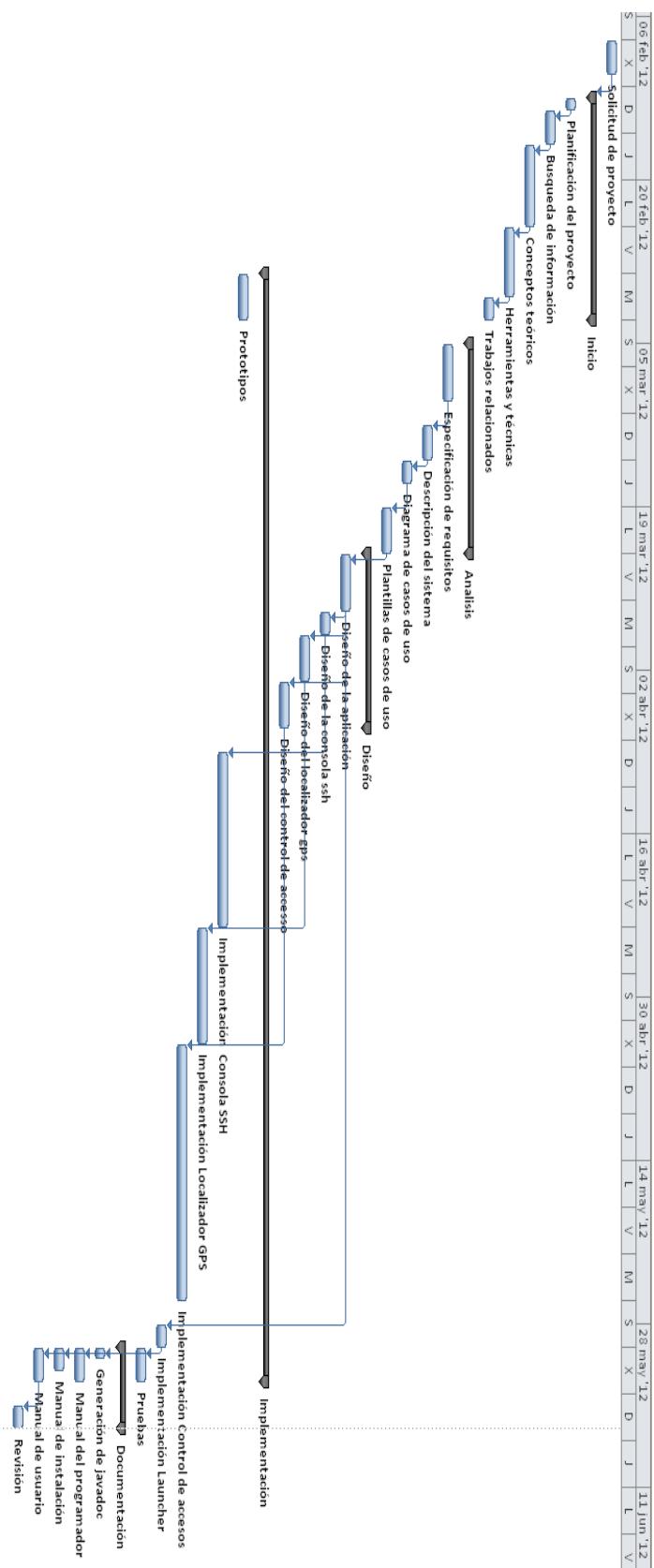


Ilustración 4: Diagrama de Gantt de la planificación inicial

### 2.1.7 Hoja de tareas de la planificación inicial

	Task Name	Duración	Comienzo	Fin
1	Solicitud de proyecto	3 días	mié 08/02/12	vie 10/02/12
2	■ Inicio	15 días	lun 13/02/12	vie 02/03/12
3	Planificación del proyecto	1 día	lun 13/02/12	lun 13/02/12
4	Busqueda de información	3 días	mar 14/02/12	jue 16/02/12
5	Conceptos teóricos	5 días	vie 17/02/12	jue 23/02/12
6	Herramientas y técnicas	4 días	vie 24/02/12	mié 29/02/12
7	Trabajos relacionados	2 días	jue 01/03/12	vie 02/03/12
8	■ Análisis	14 días	lun 05/03/12	jue 22/03/12
9	Especificación de requisitos	5 días	lun 05/03/12	vie 09/03/12
10	Descripción del sistema	3 días	lun 12/03/12	mié 14/03/12
11	Diagrama de casos de uso	2 días	jue 15/03/12	vie 16/03/12
12	Plantillas de casos de uso	4 días	lun 19/03/12	jue 22/03/12
13	■ Diseño	11 días	vie 23/03/12	vie 06/04/12
14	Diseño de la aplicación	3 días	vie 23/03/12	mar 27/03/12
15	Diseño de la consola ssh	2 días	mié 28/03/12	jue 29/03/12
16	Diseño del localizador gps	2 días	vie 30/03/12	lun 02/04/12
17	Diseño del control de acceso	4 días	mar 03/04/12	vie 06/04/12
18	■ Implementación	69 días	mar 28/02/12	vie 01/06/12
19	Prototipos	4 días	mar 28/02/12	vie 02/03/12
20	Implementación Consola SSH	11 días	lun 09/04/12	lun 23/04/12
21	Implementación Localizador GPS	8 días	mar 24/04/12	jue 03/05/12
22	Implementación Control de accesos	16 días	vie 04/05/12	vie 25/05/12
23	Implementación Launcher	2 días	lun 28/05/12	mar 29/05/12
24	Pruebas	3 días	mié 30/05/12	vie 01/06/12
25	■ Documentación	5 días	mié 30/05/12	mar 05/06/12
26	Generación de javadoc	1 día	mié 30/05/12	mié 30/05/12
27	Manual del programador	3 días	mié 30/05/12	vie 01/06/12
28	Manual de instalación	2 días	mié 30/05/12	jue 31/05/12
29	Manual de usuario	3 días	mié 30/05/12	vie 01/06/12
30	Revisión	2 días	lun 04/06/12	mar 05/06/12

Ilustración 5: Tareas de la planificación inicial

## 2.2 Planificación final

La planificación final engloba todas las tareas realizadas en todas las fases del proyecto, se diferencia con el de planificación inicial, en que en este se muestra los trabajos y tiempos empleados en realizar las tareas, mientras en el inicial eran estimaciones.

En nuestro caso ha habido un retraso en la finalización del proyecto, provocados por emplear más tiempo del estimado en algunas tareas (Conceptos teóricos, trabajos relacionados, documentación) y por el parón para estudiar exámenes en la segunda mitad de Mayo.

### 2.2.1 Diagrama de Gant de la planificación final

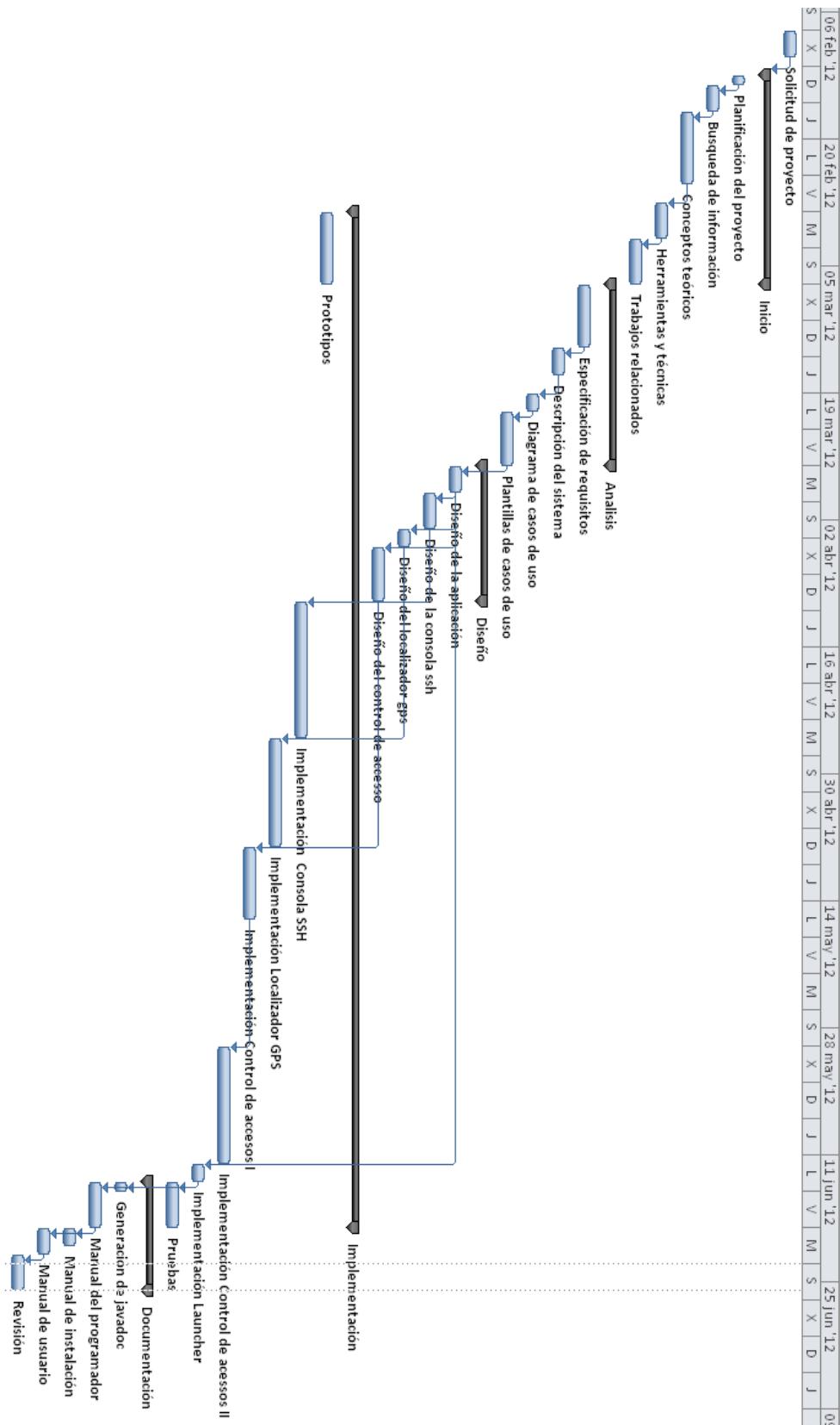


Ilustración 6: Diagrama de Gant de la planificación final

## 2.2.2 Hoja de tareas de la planificación final

	Task Name	Duración	Comienzo	Fin
1	Solicitud de proyecto	3 días	mié 08/02/12	vie 10/02/12
2	▣ Inicio	17 días	lun 13/02/12	mar 06/03/12
3	Planificación del proyecto	1 día	lun 13/02/12	lun 13/02/12
4	Busqueda de información	3 días	mar 14/02/12	jue 16/02/12
5	Conceptos teóricos	6 días	vie 17/02/12	vie 24/02/12
6	Herramientas y técnicas	4 días	lun 27/02/12	jue 01/03/12
7	Trabajos relacionados	3 días	vie 02/03/12	mar 06/03/12
8	▣ Análisis	14 días	mié 07/03/12	lun 26/03/12
9	Especificación de requisitos	5 días	mié 07/03/12	mar 13/03/12
10	Descripción del sistema	3 días	mié 14/03/12	vie 16/03/12
11	Diagrama de casos de uso	2 días	lun 19/03/12	mar 20/03/12
12	Plantillas de casos de uso	4 días	mié 21/03/12	lun 26/03/12
13	▣ Diseño	11 días	mar 27/03/12	mar 10/04/12
14	Diseño de la aplicación	3 días	mar 27/03/12	jue 29/03/12
15	Diseño de la consola ssh	2 días	vie 30/03/12	lun 02/04/12
16	Diseño del localizador gps	2 días	mar 03/04/12	mié 04/04/12
17	Diseño del control de acceso	4 días	jue 05/04/12	mar 10/04/12
18	▣ Implementación	80 días	mar 28/02/12	lun 18/06/12
19	Prototipos	6 días	mar 28/02/12	mar 06/03/12
20	Implementación Consola SSH	11 días	mié 11/04/12	mié 25/04/12
21	Implementación Localizador GPS	8 días	jue 26/04/12	lun 07/05/12
22	Implementación Control de accesos I	6 días	mar 08/05/12	mar 15/05/12
23	Implementación Control de accesos II	9 días	mié 30/05/12	lun 11/06/12
24	Implementación Launcher	2 días	mar 12/06/12	mié 13/06/12
25	Pruebas	3 días	jue 14/06/12	lun 18/06/12
26	▣ Documentación	8 días	jue 14/06/12	lun 25/06/12
27	Generación de javadoc	1 día	jue 14/06/12	jue 14/06/12
28	Manual del programador	3 días	jue 14/06/12	lun 18/06/12
29	Manual de instalación	2 días	mar 19/06/12	mié 20/06/12
30	Manual de usuario	3 días	mar 19/06/12	jue 21/06/12
31	Revisión	2 días	vie 22/06/12	lun 25/06/12

Ilustración 7: Tareas de la planificación final

## 3 ESTUDIO DE VIABILIDAD

### 3.1 Viabilidad económica

En este apartado se describe el análisis económico del proyecto que señalará los costes de desarrollo del proyecto. En términos económicos se enfrenta lo que se produce con lo que se gasta. Para que el proyecto sea totalmente viable debe de cumplir con los

objetivos que se establecen y que sea a coste eficiente.

### 3.1.1 Costes de personal

El proyecto será realizado por dos desarrolladores, encargados de realizar todas las tareas de investigación, planificación, análisis, diseño, implementación y documentación.

La duración del proyecto se estima en 98 días trabajando una media de 4 horas al día por cada desarrollador.

Se estima una remuneración de 12 €/hora trabajada.

- **Coste salarial**

$$98 \text{ días} \times 4 \text{ h/día} = 392 \text{ horas trabajadas}$$

$$392 \text{ horas} \times 12 \text{ €/hora} = 4.704 \text{ € por desarrollador}$$

Al ser 2 desarrolladores el coste total de personal sera:

$$4.704 \text{ €} \times 2 = 9.408 \text{ €}$$

- **Costes de seguridad social**

Según la ley de Seguridad Social de los sueldos que se paga a los trabajadores, un porcentaje debe ser abonado al Estado como pago de impuestos. La formula para calcular los gastos de personal es la siguiente:

$$\textit{Gastos de personal al mes} = \textit{Sueldo bruto} + \textit{Seguridad Social}$$

Según la tabla de bases de cotización de 2012, los desarrolladores pertenecen al grupo de cotización 2 (Ingenieros Técnicos, Peritos y Ayudantes Titulados), con una base mínima de cotización de 867,00 €/mes y una base máxima de 3262,50 €/mes.

Sobre esa base se aplican diferentes tipos de cotización, pero se aplicarán las contingencias comunes, según esto la empresa paga un 23,6 % y el trabajador un 4,7 %, haciendo un total de 28,3 %.

$$4704 \text{ €} \times 28,3 \% = 1331,23 \text{ €}$$

<b>Coste de personal</b>	<b>Desarrollador 1</b>	<b>Desarrollador 2</b>
Sueldo Bruto	4.704,00 €	4.704,00 €
Seguridad Social	1.331,23 €	1.331,23 €
Total	6.035,23 €	6.035,23 €
<b>Gasto Total</b>	<b>12.070,46 €</b>	

Tabla 2: Costes de personal

Finalmente, el coste de personal asciende a: **12.070,46 €**.

### 3.1.2 Costes de hardware

Para poder realizar todas las fases de desarrollo, cada programador necesitará usar un ordenador, ademas de un dispositivo móvil donde poder probar funcionalidades que no se puede probar en el emulador.

Se estima el valor de cada ordenador en 900 € y el dispositivo móvil en 350 €. En este caso no se considerará el importe total de estos dado que se podrán seguir utilizando en un futuro para la realización de otros proyectos, por lo que se procederá a calcular la amortización de estos costes.

Según la ley de Impuestos de Sociedades, la vida media del inmovilizado de hardware está entre los 4 y 9 años. En nuestro caso y debido a que los desarrolladores de software tiende a renovar los equipos con mayor facilidad, consideraremos la vida media como 4 años, por tanto la amortización sera:

- Periodo de amortización: 4 años = 48 meses.
- Precio de los equipos: 900 €/equipo x 2 equipos + 350 €/dispositivo móvil = 2150 €
- Coste de amortización:  $2150 \text{ €} / 48 \text{ meses} = 44,79 \text{ €/mes}$

Como el proyecto esta estimado que tarde 4 meses, el coste de amortización de hardware será:

$$44,79 \text{ €/mes} \times 4 \text{ meses} = 179,16 \text{ €}$$

Por lo que los gastos de hardware ascienden a **179,16 €**.

### 3.1.3 Costes de software

Para realizar el proyecto se necesitá instalar distintas herramientas software en los equipos de ambos programadores, aunque la gran mayoría de las herramientas utilizadas son de software libre las incluiremos en el apartado de costes con importe cero.

<b>Software</b>	<b>Licencia</b>	<b>Cantidad</b>	<b>Coste</b>	<b>Total</b>
Microsoft Windows 7	Individual	2	199,00 €	399,98 €
Ubuntu Server 10.04 LTS	GPL	1	0,00 €	0,00 €
Java JDK 6 u33	GNU GPL	3	0,00 €	0,00 €
Android SDK rev18	ASL 2.0	2	0,00 €	0,00 €
Eclipse 1.4.2	Open Source	2	0,00 €	0,00 €
SQLite	Public Domain	2	0,00 €	0,00 €
MySQL 5.0	GNU GLP v2	1	0,00 €	0,00 €
Apache Tomcat 7.0	ASL 2.0	1	0,00 €	0,00 €
Apache Axis2 1.6.2	ASL 2.0	1	0,00 €	0,00 €
JSVC 1.0.10	ASL 2.0	1	0,00 €	0,00 €
JSCH	BSD	2	0,00 €	0,00 €
Open Office 3.3.0	Open Source	2	0,00 €	0,00 €
Git	GNU GPL v2	2	0,00 €	0,00 €
TortoiseGit	GNU GPL	2	0,00 €	0,00 €
Astah Community	Versión de prueba	2	0,00 €	0,00 €
Jmeter 2.7	ASL 2.0	1	0,00 €	0,00 €
Droid Explorer 0.8.8.3	MPL	1	0,00 €	0,00 €
Microsoft Project 2010	Versión de prueba	1	0,00 €	0,00 €
<b>Coste Total</b>				<b>399,98 €</b>

Tabla 3: Costes de software

Como ocurría con los costes de hardware, los costes de software tambien son amortizables y según a Ley de Impuestos de Sociedades, el tiempo de vida medio para el software se encunetra entre los 3 y 6 años.

En nuestro caso como tan solo incurrimos en gastos de sistema operativo, por lo que estimamos para ellos una vida media de 5 años, por lo cual:

- Periodo de amortización: 5 años = 60 meses.
- Precio de los equipos: 399,98 €
- Coste de amortización:  $399,98 \text{ €} / 60 \text{ meses} = 6,66 \text{ €/mes}$

Como el proyecto esta estimado que tarde 4 meses, el coste de amortización de software será:

$$6,66 \text{ €/mes} \times 4 \text{ meses} = 26,64 \text{ €}$$

Por lo que los gastos de software ascienden a **26,64 €**.

### 3.1.4 Costes generales

A parte de los costes anteriormente descritos se debe contar con otros gastos generales derivados de la actividad que se va a realizar y el emplazamiento donde realizará la actividad.

Tipo de gasto	€/mes	Meses	Total
Alquiler de oficina	400,00 €	4	1.600,00 €
Material de oficina	20,00 €	4	80,00 €
Electricidad	30,00 €	4	120,00 €
Internet	25,00 €	4	100,00 €
Impresión y encuadernación del proyecto			150,00 €
<b>Coste Total</b>			<b>2.050,00 €</b>

Tabla 4: Costes generales

Por lo que los costes generales suben a **2.050,00 €**.

### 3.1.5 Costes totales

Para calcular el coste total del proyecto basta con sumar los costes calculados anteriormente.

Coste	Importe
Coste de personal	12.070,46 €
Coste de hardware	179,16 €
Coste de software	26,64 €
Costes generales	2.050,00 €
<b>Coste Total</b>	<b>14.326,26 €</b>

Tabla 5: Coste total

### 3.1.6 Análisis de costes-beneficios

Se proponen dos versiones de la aplicación para ser distribuidas, una gratuita con funcionalidades reducidas, y otra de pago con todas las funcionalidades disponibles.

La versión gratuita tendría la restricción de solo poder gestionar un solo servidor a la vez, y la versión de pago podría gestionar todos los que quisiera el usuario.

Estimando un precio para la aplicación de pago de 1,99 €, se calcula el número de licencias mínimas a vender para empezar a obtener beneficios.

$$14.326,26 \text{ €} / 1,99 \text{ €} = 7200$$

Se necesitan vender **7200** aplicaciones a 1,99€ para empezar a obtener beneficios.

### 3.2 Viabilidad legal

Para poder estudiar la viabilidad legal de nuestro proyecto es necesario saber qué es una licencia software, ya que de lo que se trata cuando se realiza el análisis de viabilidad legal es determinar bajo qué tipo de licencia será distribuida nuestra aplicación.

Una licencia software es un contrato entre el autor o distribuidor de la aplicación informática y el usuario del programa, que sirve para que el producto software se utilice según una serie de términos y condiciones que vienen establecidos en las cláusulas del contrato.

Se propone la distribución de la aplicación bajo una licencia de tipo de usuario final (EULA o End User License Agreement). Se ha comprobado que las licencias de las bibliotecas de otros de las que se hace uso, nos permiten este uso de esta licencia, y vemos que es la más adecuada para nuestra aplicación.

Una licencia EULA es un contrato entre el fabricante y/o autor y el usuario final de una aplicación, en ella se detalla cómo se puede usar el software y cualquier restricción que el fabricante imponga.

Al adquirir una Licencia de Uso Final, no se paga por la propiedad del programa, sino solo por los derechos de utilizarlo, comprometiéndose a respetar todas las condiciones que impone el propietario en la licencia. La licencia define las formas en las que se puede utilizar la aplicación. El código fuente no se libera, es de secreto comercial. Esta licencia es personal e intransferible, y es categorizado de delito el que se copie o pase a otra persona.

# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

#### **Anexo II: Especificación de requisitos**

**Alumnos**      David Herrero de la Peña  
                  Jonatan Santos Barrios

**Tutor**      Carlos López Nozal  
                  DEPARTAMENTO DE INGENIERÍA CIVIL  
                  Área de Lenguajes y Sistemas Informáticos  
                  Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## **Índice de contenido**

1. Introducción.....	58
2. Objetivos del proyecto.....	58
3. Lista de usuarios participantes.....	58
4. Descripción del sistema actual.....	58
5. Catalogo de requisitos del sistema.....	59
5.1. Requisitos funcionales.....	59
5.1.1. Requisitos funcionales módulo control de accesos.....	59
5.1.2. Requisitos funcionales módulo teminal SSH.....	61
5.1.3. Requisitos funcionales módulo localización GPS.....	62
5.2. Requisitos de información.....	63
5.2.1. Web Service.....	63
5.2.2. Módulo terminal SSH.....	64
5.2.3. Módulo control de accesos.....	65
5.2.4. Módulo localizador GPS.....	67
5.3. Requisitos no funcionales.....	67
6. Especificaciones de requisitos.....	68
6.1. Diagrama de casos de uso.....	68
6.2. Casos de uso.....	69
7. Interfaz de usuario.....	77
8. Pruebas.....	80

## Índice de ilustraciones

Ilustración 8: Módelo Entidad-Relación Web Service.....	64
Ilustración 9: Módelo Entidad-Relación Scripts.....	64
Ilustración 10: Módelo Entidad-Relación Servidores.....	65
Ilustración 11: Módelo Entidad-Relación Email.....	66
Ilustración 12: Módelo Entidad-Relación Servidores.....	66
Ilustración 13: Módelo Entidad-Relación Coordenadas.....	67
Ilustración 14: Diagrama de casos de uso.....	68
Ilustración 15: Diagrama de secuencia de una conexión SSH.....	71
Ilustración 16: Diagrama de secuencia de activación del localizador GPS.....	73
Ilustración 17: Diagrama de secuencia del control de accesos.....	75
Ilustración 18: Pantalla principal.....	78
Ilustración 19: Pantalla de opciones.....	79
Ilustración 20: Pantalla con pestañas.....	80

## Índice de tablas

Tabla 6: Plantilla de requisitos.....	59
Tabla 7: RF01 – Obtención de archivos log.....	59
Tabla 8: RF02 – Análisis de los archivos log.....	60
Tabla 9: RF03 – Escucha de peticiones de información.....	60
Tabla 10: RF04 – Control de notificaciones descargadas en dispositivo.....	60
Tabla 11: RF05 – Borrar notificaciones para dispositivos.....	60
Tabla 12: RF06 – Notificaciones de acceso.....	61
Tabla 13: RF07 – Comunicación SSH.....	61
Tabla 14: RF08 – Servidores recientes.....	61
Tabla 15: RF09– Almacenamiento de comandos.....	61
Tabla 16: RF10– Creación de scripts.....	62
Tabla 17: RF11 – Envío de coordenadas.....	62
Tabla 18: RF12 – Almacenamiento de localizaciones.....	62
Tabla 19: RF13 – Ejecución en background.....	62
Tabla 20: RF14 – Activación remota.....	63
Tabla 21: RF15 – Borrado de datos.....	63
Tabla 22: RF16 – Envío instantáneo.....	63
Tabla 23: Plantilla de Casos de Uso.....	69
Tabla 24: Caso de Uso Consola Android SSH.....	70
Tabla 25: Caso de Uso Localización GPS.....	72
Tabla 26: Caso de Uso Control de Accesos a Servicios.....	74
Tabla 27: Caso de Uso Gestión de Filtros.....	76
Tabla 28: Caso de Uso Gestión de Filtros.....	77
Tabla 29: Pruebas Control de Accesos.....	81
Tabla 30: Pruebas Terminal SSH.....	81
Tabla 31: Pruebas Localizador GPS.....	82

## **LISTA DE CAMBIOS**

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>
1.0	28/03/12	Primera versión: Introducción, Objetivos del proyecto, Lista de usuarios participantes.
2.0	30/03/12	Segunda versión: Catalogo de requisitos del sistema.
3.0	02/04/12	Tercera versión: Casos de uso.
4.0	09/06/12	Cuarta versión: Interfaz de usuario, Pruebas.

## 1 INTRODUCCIÓN

Este anexo de especificaciones de requisitos de software cumple con dos funciones básicas.

En primer lugar debe recoger los requisitos del sistema software, atendiendo a las necesidades indicadas por el cliente, de manera que sirva como elemento contractual entre el cliente y los desarrolladores. En segundo lugar especificará dichos requisitos representándolos de forma gráfica y textual, presentando los distintos enfoques del sistema.

Para ellos se creara un catálogo de requisitos, para indicar de forma general los principales requerimientos, junto con las funciones y restricciones del sistema, y a continuación se detallaran los requisitos utilizando herramientas UML como diagramas y plantillas de casos de uso.

## 2 OBJETIVOS DEL PROYECTO

A continuación se detallan los objetivos relacionados con el funcionamiento del sistema:

La principal meta es la creación de una aplicación para dispositivo móvil Android, que permita la gestión de distintas partes de un servidor, para ello la aplicación contara con tres módulos:

- Módulo Control de Accesos: El objetivo de este módulo es la monitorización en tiempo real del control de accesos de usuarios a un servidor, tanto si el intento de acceso ha tenido éxito como si no, además del control del servidor de correo.
- Módulo Terminal SSH : Este módulo tiene como objetivo la simulación de una terminal de comandos del servidor mediante una conexión segura SSH.
- Módulo Localización GPS: Este módulo tiene como objetivo la localización y seguimiento de un dispositivo Android en caso de pérdida.

## 3 LISTA DE USUARIOS PARTICIPANTES

Para establecer los requisitos de un proyecto es necesario reunirse con el cliente que indicara las necesidades y lo que espera de la aplicación.

El cliente al que va dirigido la realización de este proyecto es Rubén Arribas Barrio responsable de las empresas expertosenandroid.com y expertosenweb.com.

Uno de los desarrolladores y colaborador del proyecto es Carlos López Nozal, que es quien nos puso en contacto con el cliente y actuó de intermediario.

## 4 DESCRIPCIÓN DEL SISTEMA ACTUAL

En este proyecto no se parte de ningún sistema ya implantado, si no que se se empieza a trabajar desde cero en todos los aspectos, por lo que no es posible facilitar información al respecto en este apartado.

## 5 CATALOGO DE REQUISITOS DEL SISTEMA

En este apartado se especifican todos los requisitos funcionales del sistema.

### 5.1 Requisitos funcionales

Para la descripción de los requisitos se empleara la siguiente plantilla:

<b>Identificador</b>	Nombre del requisito
<b>Versión</b>	Versión y fecha de la versión actual
<b>Descripción</b>	Descripción del requisito
<b>Importancia</b>	Importancia del requisito
<b>Comentarios</b>	Comentarios adicionales

Tabla 6: Plantilla de requisitos

- **Identificador del requisito:** hace referencia al código identificativo que representa a un único requisito.
- **Nombre del requisito:** se dará un nombre que resuma el requisito.
- **Versión:** número y fecha de la versión actual del requisito.
- **Descripción:** definirá de forma más detallada que el nombre el requisito.
- **Importancia:** indica la relevancia del requisito.
- **Comentarios:** información adicional que se considere importante añadir.

Aunque Android Server Status es una única aplicación, a continuación se muestran los requisitos funcionales de cada módulo de la aplicación.

#### 5.1.1 Requisitos funcionales módulo control de accesos

Esta parte de la aplicación tiene tres aplicaciones distintas y que para dar total funcionalidad deberán trabajar juntas, esta compuesta por un demonio o servicio que escuchara los mensajes de log, otro el servicio web que contestara a la aplicación móvil cuando esta pregunte información sobre las notificaciones y la aplicación móvil.

<b>RF01</b>	Obtención de mensajes de log.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El demonio deberá escuchar los mensajes log del sistema, a través de servicio syslog.
<b>Importancia</b>	Alta
<b>Comentarios</b>	Syslog es un servicio del sistema operativo encargado de enviar, recibir y registrar los mensajes log del sistema.

Tabla 7: RF01 – Obtención de archivos log

<b>RF02</b>	Análisis y registro de los mensajes log.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El demonio analizará los mensajes de log en busca de nuevos intentos de accesos y almacenar en base de datos las que deben ser notificados a los dispositivos.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

*Tabla 8: RF02 – Análisis de los archivos log*

<b>RF03</b>	Escucha de peticiones de información.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El servicio web deberá escuchar constantemente las peticiones de los distintos dispositivos que preguntan información sobre las notificaciones.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

*Tabla 9: RF03 – Escucha de peticiones de información*

<b>RF04</b>	Control de notificaciones descargadas en dispositivo.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El servicio web deberá llevar un control de que dispositivos se han descargado las notificaciones.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

*Tabla 10: RF04 – Control de notificaciones descargadas en dispositivo*

<b>RF05</b>	Borrar notificaciones para dispositivos.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El servicio web permitirá borrar las notificaciones para el dispositivo que lo indique.
<b>Importancia</b>	Media
<b>Comentarios</b>	

*Tabla 11: RF05 – Borrar notificaciones para dispositivos*

<b>RF06</b>	Notificaciones de acceso.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación deberá comprobar periódicamente preguntando al servicio web si hay notificaciones nuevas en el servicio web.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla 12: RF06 – Notificaciones de acceso

### 5.1.2 Requisitos funcionales módulo terminal SSH

<b>RF07</b>	Comunicación ssh.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	Toda la comunicación con el servidor deberá realizarse usando conexiones seguras tipo ssh.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla 13: RF07 – Comunicación SSH

<b>RF08</b>	Servidores recientes.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	Se deben poder guardar los datos de acceso a los servidores a los que se acceda de forma más periódica.
<b>Importancia</b>	Media
<b>Comentarios</b>	

Tabla 14: RF08 – Servidores recientes

<b>RF09</b>	Almacenamiento de comandos.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	Se deben poder almacenar los comandos introducido en una sesión para usos posteriores.
<b>Importancia</b>	Media
<b>Comentarios</b>	

Tabla 15: RF09 – Almacenamiento de comandos

<b>RF10</b>	Creación de scripts.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	El usuario podrá usar comandos almacenados anteriormente para crear nuevos scripts.
<b>Importancia</b>	Baja
<b>Comentarios</b>	

Tabla 16: RF10– Creación de scripts

### 5.1.3 Requisitos funcionales módulo localización GPS

<b>RF11</b>	Envío de coordenadas.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación deberá tener comunicación periódica siempre que haya sido activada y le resulte posible, proporcionando el ID del teléfono y las coordenadas GPS más recientes.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla 17: RF11 – Envío de coordenadas

<b>RF12</b>	Almacenamiento de localizaciones.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación irá almacenando las sucesivas localizaciones junto con la fecha y hora en que fueron recibidas.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla 18: RF12 – Almacenamiento de localizaciones

<b>RF13</b>	Ejecución en background.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación deberá ejecutarse como servicio con el fin de garantizar que se mantenga el mayor tiempo posible en ejecución.
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla 19: RF13 – Ejecución en background

<b>RF14</b>	Activación remota.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación debe poder activarse de forma remota.
<b>Importancia</b>	Alta
<b>Comentarios</b>	Se intenta tener un pequeño control de la aplicación en caso de robos y/o pérdidas.

Tabla 20: RF14 – Activación remota

<b>RF15</b>	Borrado de datos.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación debe permitir el borrado de los datos almacenados en el teléfono en caso de urgencia.
<b>Importancia</b>	Media
<b>Comentarios</b>	Los datos que serán borrados son los de la propia aplicación.

Tabla 21: RF15 – Borrado de datos

<b>RF16</b>	Envío instantáneo.
<b>Versión</b>	1.0 – Marzo 2012
<b>Descripción</b>	La aplicación debe ser capaz de enviar los datos de localización al usuario en el menor tiempo posible.
<b>Importancia</b>	Media
<b>Comentarios</b>	

Tabla 22: RF16 – Envío instantáneo

## 5.2 Requisitos de información

### 5.2.1 Web Service

Para que la aplicación y el Web Service puedan trabajar y registrar notificaciones el servidor debe tener instalado ssh y postfix, que son servicios para conexión remota y correo electrónico respectivamente, y hacerse uso de ellos.

La información necesaria para que la aplicación pueda ser notificada se guarda en una base de datos con la siguiente estructura.

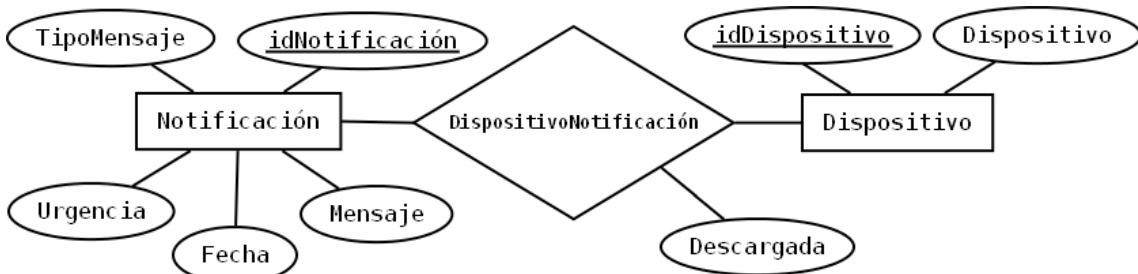


Ilustración 8: Móodelo Entidad-Relación Web Service

**Notificación:** mensaje de log recibido y que debe de ser notificado.

IdNotificación: identificador único de la notificación.

TipoMensaje: tipo de mensaje, puede ser de ssh o de correo electrónico.

Urgencia: la urgencia de la notificación, puede ser grave, media o información.

Fecha: fecha en la que se ha producido la notificación.

Mensaje: mensaje explicativo de la causa de la notificación.

**Dispositivo:** dispositivo móvil que se ha conectado para recibir notificaciones.

IdDispositivo: identificador único de dispositivo único.

Dispositivo: dirección mac del dispositivo.

**DispositivoNotificación:** relación n:m, para relacionar todas las notificaciones con los dispositivos.

Descargada: indica si la notificación ha sido descargada para cada dispositivo.

### 5.2.2 Módulo terminal SSH

Los datos necesarios a almacenar en el dispositivo móvil son los scripts y los servidores favoritos.

El modelo E-R de la base de datos es el siguiente:

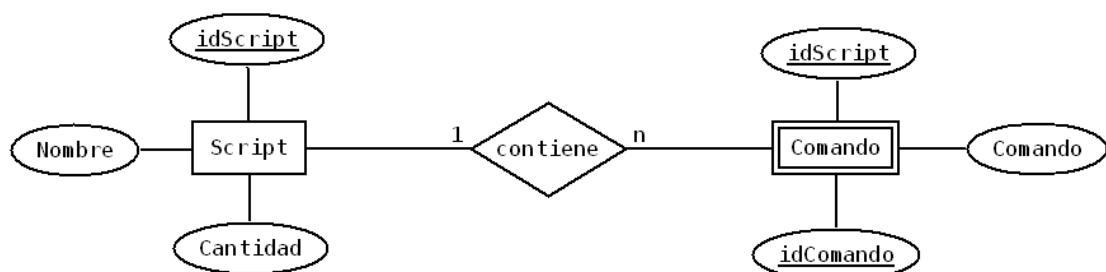


Ilustración 9: Móodelo Entidad-Relación Scripts

**Script:**

idScript : identificador del script.

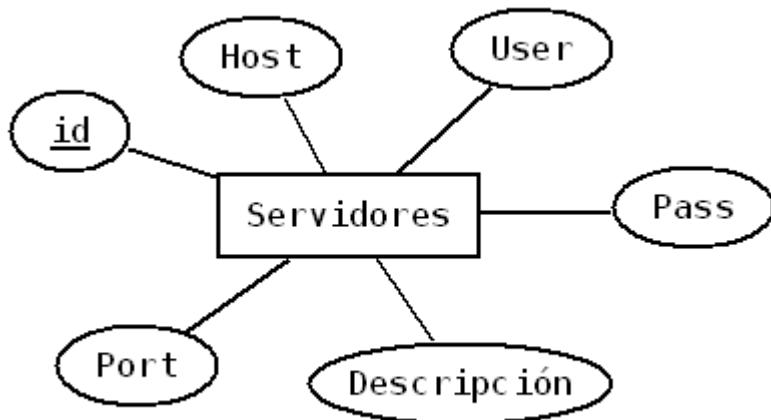
Cantidad : cantidad de comandos que contiene el script.

Nombre : nombre del script.

**Comando:**

idScript : script al que pertenece el comando.

IdComando: id del comando.



*Ilustración 10: Módelo Entidad-Relación Servidores*

Comando : texto con el comando.

**Servidores:**

id: identificador del servidor.

Host: ip del servidor.

User : usuario ssh.

Pass : contraseña del usuario.

Descripción : breve descripción.

Port : puerto al que conectarse, normalmente 22.

### 5.2.3 Módulo control de accesos

Los datos necesarios a guardar en el dispositivo móvil son los servidores a los que controlar el acceso ssh y los envíos de email.

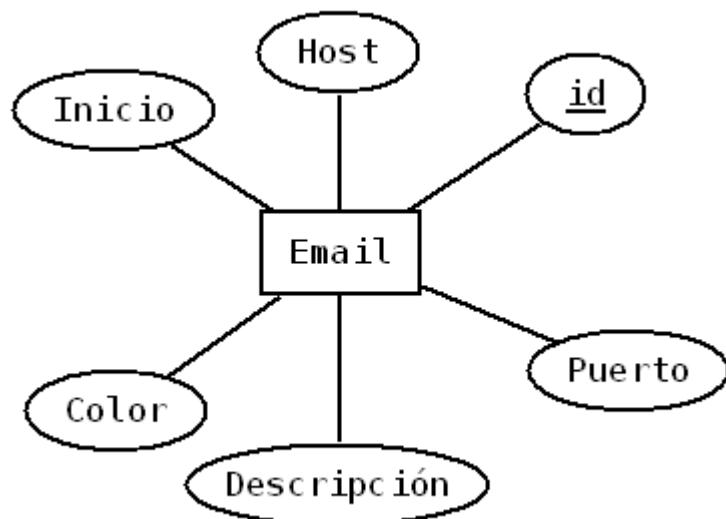


Ilustración 11: Móvilidad Entidad-Relación Email

**Email:**

Host: ip del servidor al que conectarse.

Id : identificador del servidor email.

Puerto : puerto en el que escucha el Web Service.

Descripción : breve descripción.

Color : color identificativo.

Inicio : indica si se inicia automáticamente.

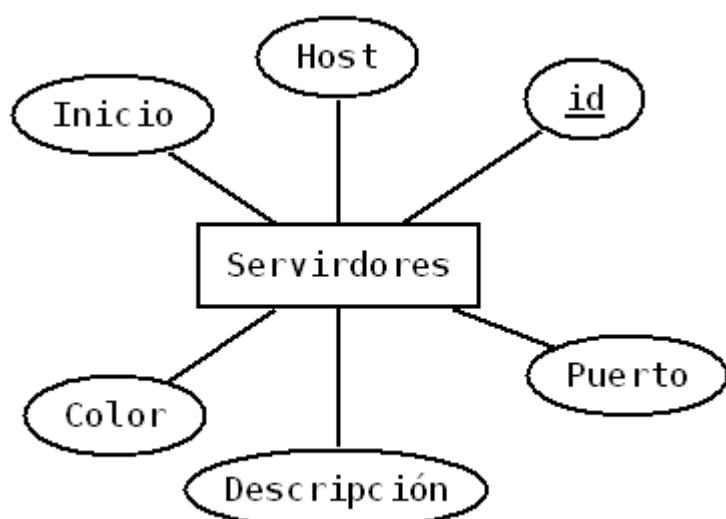


Ilustración 12: Móvilidad Entidad-Relación Servidores

**Servidores:**

Host: ip del servidor al que conectarse.

Id : identificador del servidor email.

Puerto : puerto en el que escucha el Web Service.

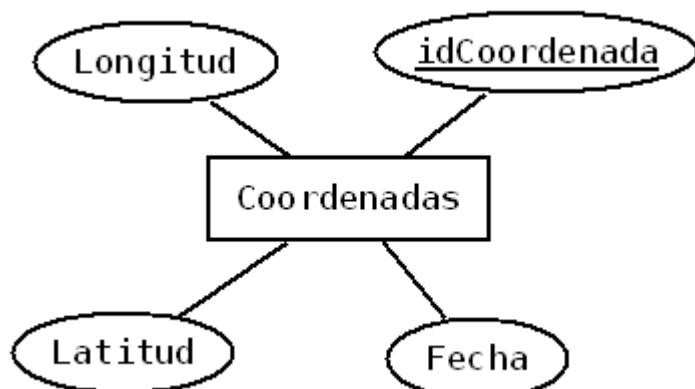
Descripción : breve descripción.

Color : color identificativo.

Inicio : indica si se inicia automáticamente.

**5.2.4 Módulo localizador GPS**

Los datos necesarios a guardar en el dispositivo móvil serán las distintas coordenadas que se van obteniendo:



*Ilustración 13: Módelo Entidad-Relación Coordenadas*

**Coordenadas:**

idCoordenada: identificador único de la corrdena.

Fecha : fecha en la que se obtuvo esa coordenada.

Longitud: longitud.

Latitud : latitud.

**5.3 Requisitos no funcionales**

- La aplicación debe ejecutarse en un dispositivo Android 2.3 o superior.
- El dispositivo móvil debe disponer de conexión de datos permanente y antena integrada GPS.
- La aplicación debe implementar algún mecanismo de seguridad para proteger los datos que se almacenan en el teléfono.
- La aplicación debe ser sencilla de utilizar e intuitiva.
- La aplicación debe ser lo más estable posible y recuperarse frente a errores inesperados.

## 6 ESPECIFICACIONES DE REQUISITOS

### 6.1 Diagrama de casos de uso

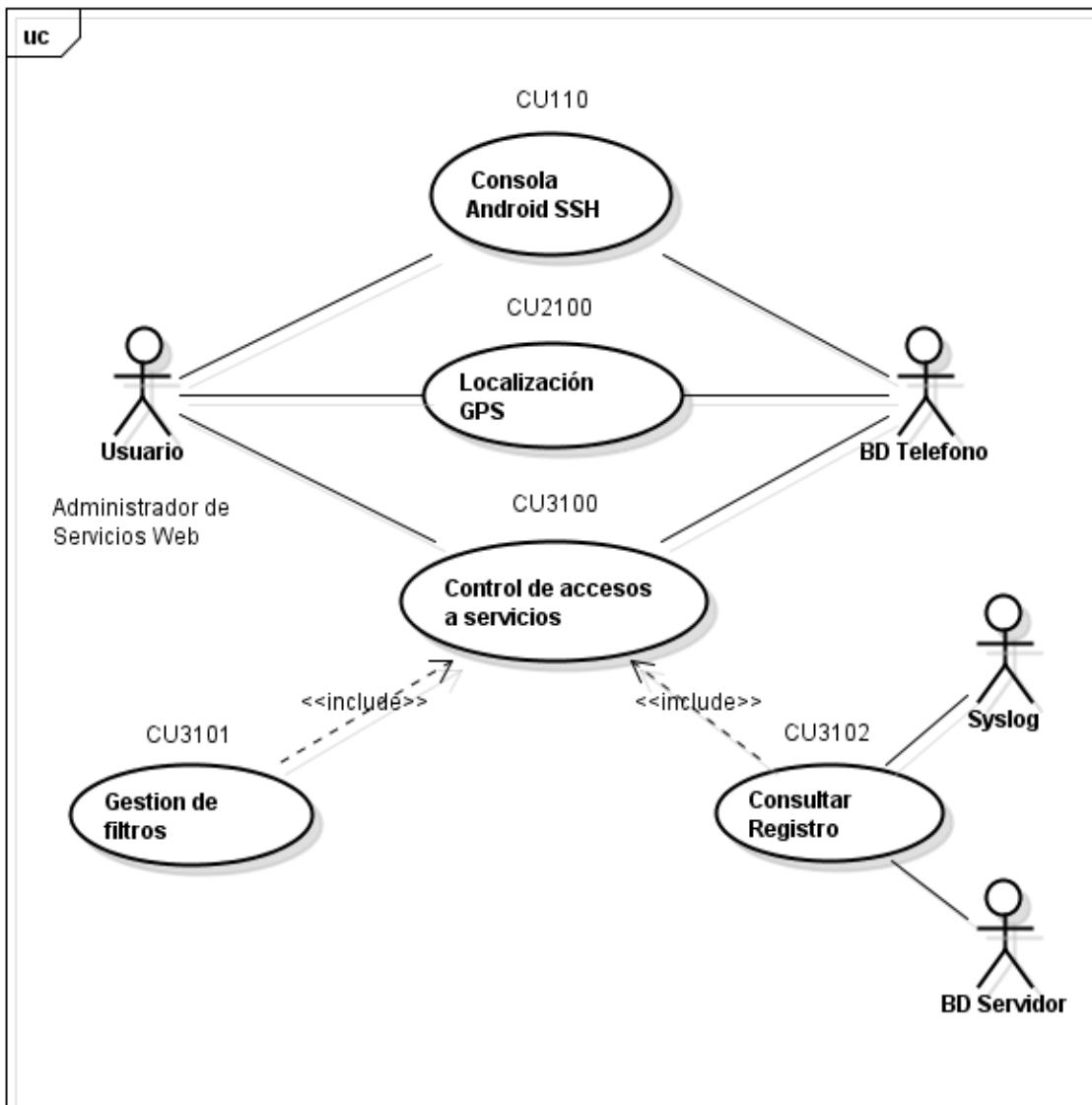


Ilustración 14: Diagrama de casos de uso

## 6.2 Casos de uso

Se usará esta plantilla para definir los casos de uso.

<b>Identificador</b>	Caso de Uso	
<b>Versión</b>	Versión y Fecha	
<b>Requisitos Asociados</b>	<ul style="list-style-type: none"> <li>Lista de requisitos</li> </ul>	
<b>Descripción</b>	Descripción del caso e Uso	
<b>Precondición</b>	Precondiciones	
<b>Acciones</b>	<b>Paso</b>	<b>Acción</b>
	p1	{El actor , el sistema}<acción/es realizada/s por actor/sistema>
	p2	Se realiza el caso de uso <caso de uso>
	p3	Si <condición>, {el actor <actor>,el sistema} <acción/es realizada/s por actor/sistema>
	p4	Si <condición>, se realiza el caso de uso <Caso de Uso (UC-x)>
	...	...
<b>Postcondición</b>	Postcondiciones	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	pi	Si <condición de excepción>, {el actor <actor>,el sistema} <acción/es realizada/s por actor/sistema>, a continuación este caso de uso {continua,termina}
	pj	Si <condición de excepción> , se realiza el caso de uso <Caso de Uso> , a continuación este caso de uso {continua,termina}
	pk	Si <condición de excepción> , este caso de uso {continua,termina}
<b>Importancia</b>	Importancia del caso de uso	
<b>Comentarios</b>	Comentarios adicionales	

Tabla 23: Plantilla de Casos de Uso

A continuación se definen mediante esta plantilla todos los casos de uso:

<b>CU1100</b>	Consola Android SSH	
<b>Versión</b>	Versión 1.0 Marzo 2012	
<b>Requisitos Asociados</b>	RF07, RF08, RF09; RF10	
<b>Descripción</b>	Conexión y gestión mediante comandos de servidor mediante una consola virtual.	
<b>Precondición</b>	<p>La conexión se debe realizar mediante protocolo SSH.            El servidor debe de ser accesible y estar funcionando.            El usuario debe de tener conocimientos de administración de servicios web.</p>	
<b>Acciones</b>	Paso	Acción
	1	Entra en la aplicación terminal.
	2	Introduce los datos para la conexión, usuario contraseña, ip del servidor y puerto de conexión.
	3	La aplicación se conecta con el servidor y muestra una consola.
	4	El usuario introduce los comandos que desea ejecutar.
	5	La aplicación envía los comandos al servidor y muestra el resultado de estos.
	6	El usuario decide si quiere crear un script con los comandos ejecutados.
<b>Postcondición</b>	Se deben almacenar los comandos ejecutados en la ultima sesión para si el usuario lo desea poder realizar un script.	
<b>Excepciones</b>	Paso	Acción
	2	Si no se introducen los datos correctamente, no se realizara la conexión y se informara de ello al usuario.
	3	Si no se consigue un canal de comunicación se cierra la consola y se informa al usuario del error.
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Ninguno	

Tabla 24: Caso de Uso Consola Android SSH

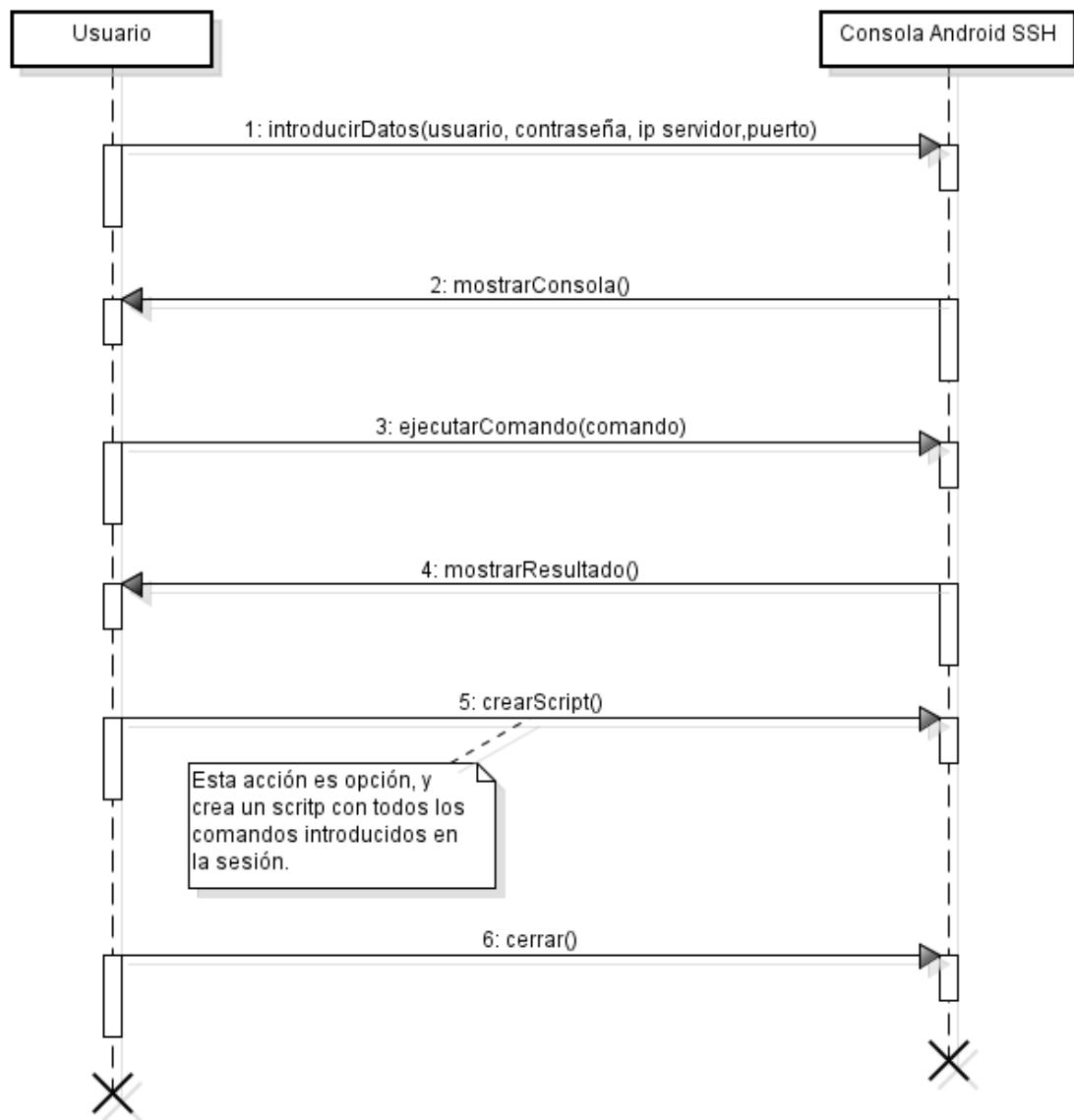


Ilustración 15: Diagrama de secuencia de una conexión SSH

<b>CU2100</b>	Localización GPS	
<b>Versión</b>	Versión 1.0 Marzo 2012	
<b>Requisitos Asociados</b>	RF11, RF12, RF13; RF14, RF15, RF16	
<b>Descripción</b>	Configuración de los datos y forma de trabajar de la aplicación GPS para su seguimiento en caso de perdida.	
<b>Precondición</b>	<p>El dispositivo debe de estar encendido y tener activada alguna de las formas que posibilitan la obtención de la posición del dispositivo (GPS, tarifa de datos o conexión por Wifi).</p> <p>El servicio debe de estar activo.</p>	
<b>Acciones</b>	Paso	Acción
	1	Entra en la aplicación localizador gps.
	2	El usuario configura los datos de textos para ser activado remotamente, o los datos de a que terminal o correo sera enviada la información..
	3	En caso de perdida el usuario desde otro móvil se enviara por mensaje la palabra clave al dispositivo para activarlo.
	4	La aplicación corre el servicio correspondiente, y manda los datos de las coordenadas al teléfono o correo configurado al inicio.
<b>Postcondición</b>	Se deben almacenar las posiciones GPS del dispositivo en la propia base de datos del dispositivo, para poder obtener una ruta de los puntos por los que ha pasado.	
<b>Excepciones</b>	Paso	Acción
	2	Si no se configuran palabras claves, el servicio no sera útil, pues no se enterara de cuando tiene que entrar en acción.
	3	Si no esta encendido el terminal, no se podrá dar aviso desde el exterior al dispositivo.
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Ninguno	

Tabla 25: Caso de Uso Localización GPS

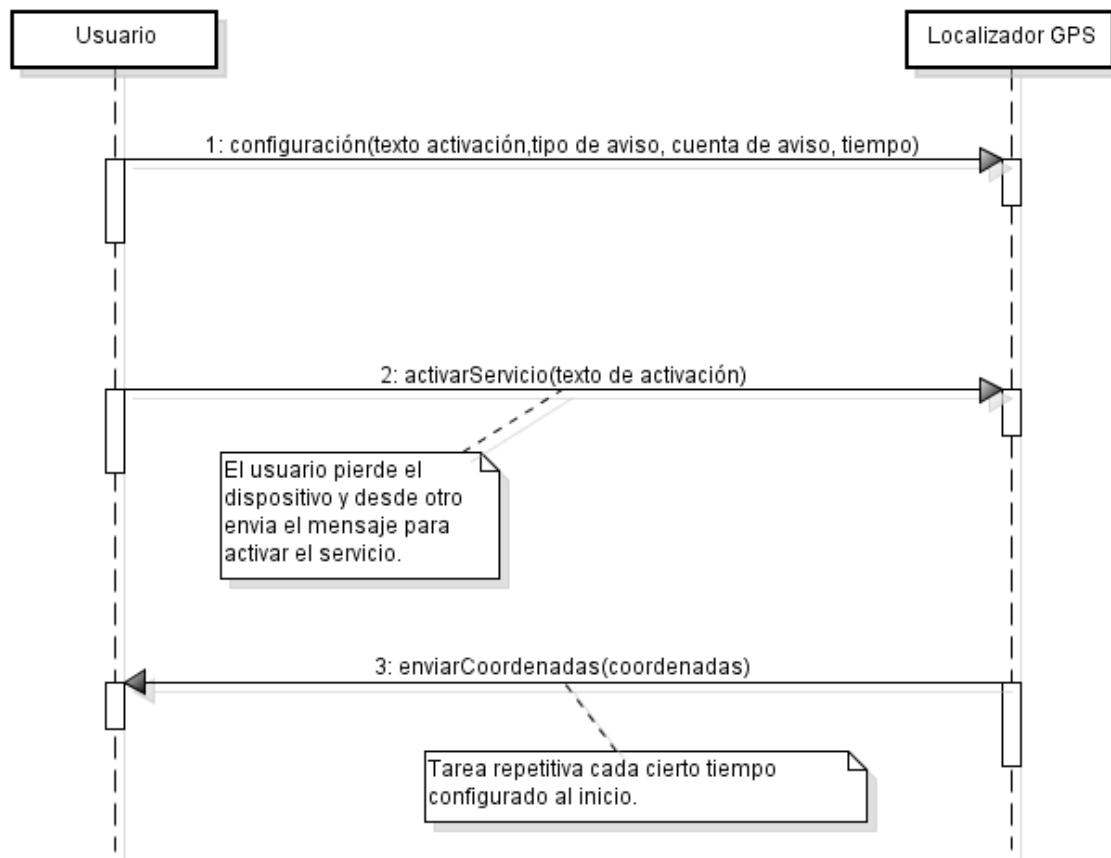


Ilustración 16: Diagrama de secuencia de activación del localizador GPS

<b>CU3100</b>	Control de acceso a servicios	
<b>Versión</b>	Versión 1.0 Marzo 2012	
<b>Requisitos Asociados</b>	RF01, RF02, RF03, RF04, RF05, RF06	
<b>Descripción</b>	Gestión de los servidores y los tipos de mensajes que el usuario quiere que le notifiquen.	
<b>Precondición</b>	El servidor debe estar arrancado y tener instalado y funcionando el correspondiente servicio web y el demonio que analice los mensajes de log.	
<b>Acciones</b>	Paso	Acción
	1	Entra en la aplicación control accesos.
	2	El usuario introduce los datos del servidor del que desea recibir las notificaciones o escoge uno usado anteriormente.
	3	El usuario activará o desactivara los avisos que desea recibir de cada servidor.
	4	La aplicación inicia o para los servicios correspondientes.
	5	La aplicación pregunta cada cierto tiempo si existen notificaciones nuevas y da un aviso si es cierto.
	6	El usuario descarga las nuevas notificaciones.
	7	El servicio web del servidor, actualiza la base de datos para grabar que la notificación ha sido descargada.
<b>Postcondición</b>	Se deben llevar un control en el servidor de que notificaciones se han descargado y para que dispositivos.	
<b>Excepciones</b>	Paso	Acción
	2	Si no se introducen los datos correctamente, no se realizara la conexión y no se recibirá ninguna notificación, se dará un aviso al usuario de que la conexión no se realizo correctamente.
	5	Si el servicio web del servidor no tiene conexión con la base de datos, no se pueden saber si existen notificaciones nuevas y se dará un aviso al usuario.
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Ninguno	

Tabla 26: Caso de Uso Control de Accesos a Servicios

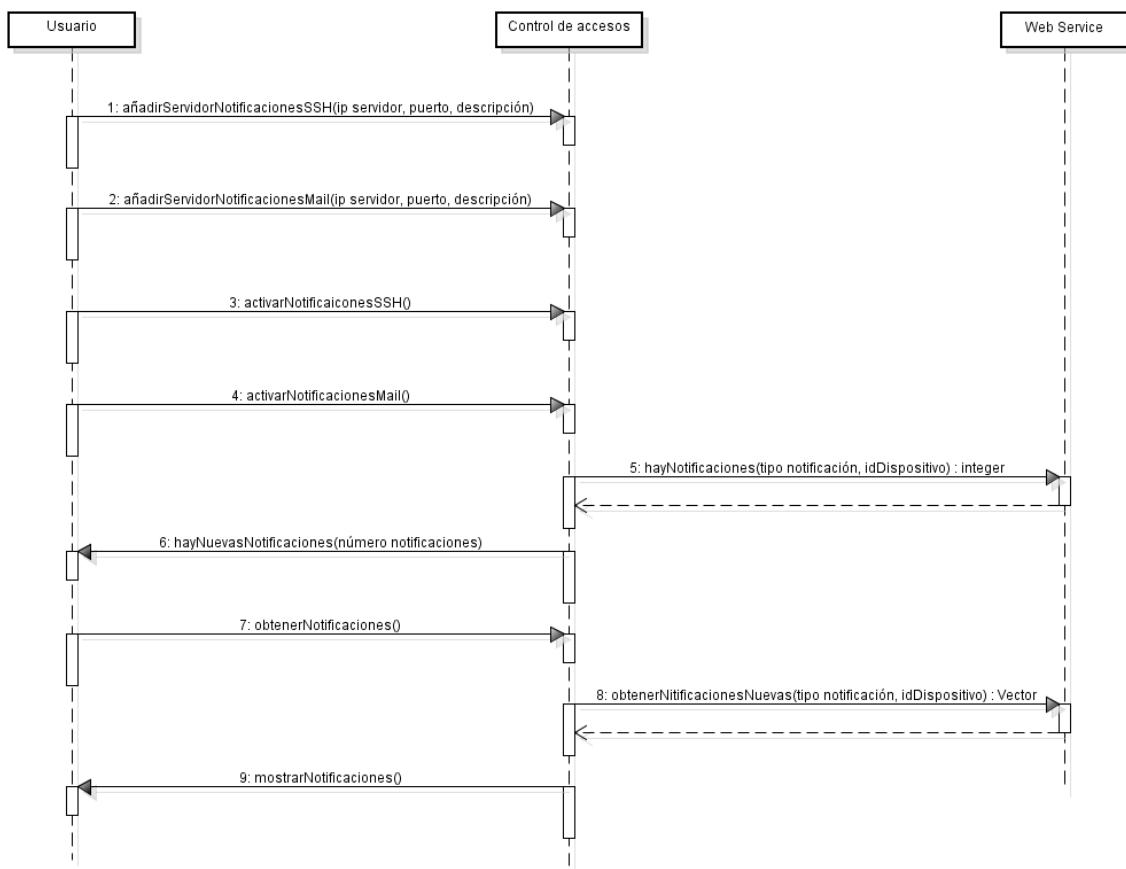


Ilustración 17: Diagrama de secuencia del control de accesos

<b>CU3101</b>	Gestión de filtros	
<b>Versión</b>	Versión 1.0 Marzo 2012	
<b>Requisitos Asociados</b>	RF01, RF02, RF03, RF04	
<b>Descripción</b>	Filtrado de las notificaciones ya descargas en el terminal, según el tipo de mensaje, la urgencia, la fecha, el servidor.	
<b>Precondición</b>	Haber descargado alguna notificación para poder realizar el filtrado	
<b>Acciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Estar en la aplicación de control de accesos y tener descargada algún notificación.
	2	El usuario se situara en una de las pestañas de la aplicación y de los desplegables elegirá el filtro que quiere aplicar
	3	La aplicación aplicara el filtro y mostrara los resultados.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si no se ha descargado notificaciones no se visualizara nada.
<b>Importancia</b>	Baja	
<b>Comentarios</b>	Ninguno	

Tabla 27: Caso de Uso Gestión de Filtros

<b>CU3102</b>	Consultar registros	
<b>Versión</b>	Versión 1.0 Marzo 2012	
<b>Requisitos Asociados</b>	RF01, RF02, RF04, RF05, RF06	
<b>Descripción</b>	Escuchar los mensajes de log provenientes del servicio syslog y grabar las notificaciones en la base de datos del servidor.	
<b>Precondición</b>	Configurar el servicio syslog para que redireccione los mensajes a nuestro demonio. Tener arrancado el demonio y que tenga conexión con la base de datos.	
<b>Acciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Arrancar el demonio junto con el sistema operativo.
	2	La aplicación escucha los mensajes.
	3	La aplicación analiza los mensajes, y registra en base de datos las notificaciones que son de su interés.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si no se ha configurado el servicio syslog no se recibirán los mensajes.
	3	Si no existe conexión con la base de datos las notificaciones no se registrarán
<b>Importancia</b>	Baja	
<b>Comentarios</b>	Ninguno	

Tabla 28: Caso de Uso Gestión de Filtros

## 7 INTERFAZ DE USUARIO

La aplicación debe estar adaptada para su uso en pantallas de resolución WVGA 800 x 480 aunque debe poder usarse en cualquier tipo de dispositivo Android.

En la interfaz de usuario se deben intentar cumplir dos requisitos básicos :

- Mostrar el máximo de información posible.
- Adaptarse al pequeño tamaño de las pantallas móviles.

Además deberá estar adaptado al uso de pantallas táctiles, en las que es necesario el uso de elementos gráficos muy representativos y sencillos de interactuar con ellos.

El apartado de vistosidad queda relegado a un segundo plano debido al tipo de público al que va dirigida la aplicación.

Como ejemplo para representar estas características se realizaron unos prototipos de interfaz gráfica, que al final fueron usados en la aplicación final :



Ilustración 18: Pantalla principal

En este ejemplo se usan botones grandes para facilitar la selección y pulsación.

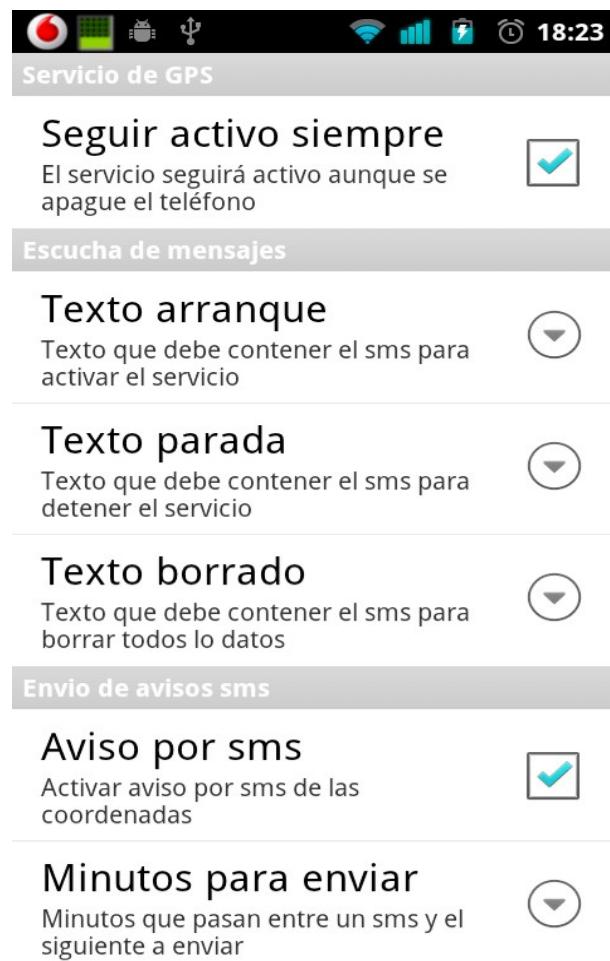


Ilustración 19: Pantalla de opciones

Las pantallas de opciones se han hecho con el mismo diseño que todas las aplicaciones Android para facilitar el manejo.



La aplicación suele usar pestañas, para representar la mayor información posible de forma ordenada, también se usan muchos las vistas tipo listas para tener los datos ordenados.

## 8 PRUEBAS

Se proponen una serie de pruebas de análisis con la entrada y salida esperadas, se dividirán en 3 apartados, uno por cada caso de uso:

<b>Pruebas de análisis : UC3 (Control de Accesos)</b>			
<b>Prueba</b>	<b>Descripción</b>	<b>Entrada</b>	<b>Salida esperada</b>
UC3_P1	No hay Internet o no se puede llegar al servidor	Datos erróneos	Se notifica el error
UC3_P2	Se cierra la aplicación por error	Ninguna	Funciona con normalidad
UC3_P3	Los puertos por los que se intenta conectar están cerrados	Puerto	Se notifica el error
UC3_P4	Se intenta descargar notificaciones de un servidor que no existe	Datos erróneos	No hace nada
UC3_P5	Se corta la conexión mientras se están descargando datos	Falta de conexión	No hace nada
UC3_P6	Se filtra por datos que no existen	Datos erróneos	No hace nada

Tabla 29: Pruebas Control de Accesos

<b>Pruebas de análisis : UC2 (Terminal SSH)</b>			
<b>Prueba</b>	<b>Descripción</b>	<b>Entrada</b>	<b>Salida esperada</b>
UC2_P1	Se introduce la dirección IP de un servidor que no permite conexión ssh	IP servidor	Se notifica error
UC2_P2	El servidor se ha quedado sin conexión a internet	IP servidor	Se notifica el error
UC2_P3	La dirección IP del servidor no existe	IP servidor	Se notifica el error
UC2_P4	El comando utilizado no existe	Comando incorrecto	Se notifica el error

Tabla 30: Pruebas Terminal SSH

<b>Pruebas de análisis : UC1 (Localizador GPS)</b>			
<b>Prueba</b>	<b>Descripción</b>	<b>Entrada</b>	<b>Salida esperada</b>
UC1_P1	No hay gps ni red 3g para obtener localizaciones	Ninguna	No se almacenan datos
UC1_P2	No hay conexión a internet para enviar los emails	Datos de gps	No se hace nada
UC1_P3	Los datos de las cuentas de correo son erróneos	Datos de configuración	No se hace nada
UC1_P4	La aplicación se cierra	Ninguna	Sigue en funcionamiento
UC1_P5	No existen localizaciones guardadas al enviar	Ninguna	Se envía un mensaje vacío o de información
UC1_P6	Se pierde la señal de gps mientras se intentaban recopilar datos	Ninguna	Sigue en funcionamiento
UC1_P7	El mensaje sms no se recibe correctamente	SMS	No se hace nada
UC1_P8	Se corta la conexión mientras se están enviando datos	Posiciones GPS	Error del sistema

Tabla 31: Pruebas Localizador GPS

# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

#### **Anexo III - Especificaciones de diseño**

Alumnos      David Herrero de la Peña  
                  Jonatan Santos Barrios

Tutor      Carlos López Nozal  
                  DEPARTAMENTO DE INGENIERÍA CIVIL  
                  Área de Lenguajes y Sistemas Informáticos  
                  Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## Índice de contenido

1. Introducción.....	91
2. Ámbito de software.....	91
3. Diseño de datos.....	92
3.1. Objetos de datos y estructuras de datos.....	92
3.1.1. Módulo Consola SSH.....	92
3.1.2. Módulo Control de Accesos.....	93
3.2. Estructuras de archivos y bases de datos.....	93
3.2.1. Web Service.....	95
3.2.1.1. Modelo lógico.....	96
3.2.1.2. Modelo físico.....	96
3.2.2. Módulo Consola.....	96
3.2.2.1. Modelo lógico.....	96
3.2.2.2. Modelo físico.....	97
3.2.3. Módulo GPS.....	97
3.2.3.1. Modelo lógico.....	97
3.2.3.2. Modelo físico.....	98
3.2.4. Módulo Control de Accesos.....	98
3.2.4.1. Modelo lógico.....	98
3.2.4.2. Modelo físico.....	98
4. Diseño arquitectónico.....	99
4.1. Estilos arquitectónicos utilizados.....	99
4.1.1. Cliente-Servidor.....	99
4.1.2. Arquitectura en capas.....	100
4.2. Patrones de diseño utilizados.....	101
4.2.1. Singleton.....	101
4.2.2. Fachada.....	101
4.3. Diseño general.....	102
4.4. Descripción detallada.....	103
4.4.1. Módulo Consola.....	104
4.4.2. Módulo GPS.....	105
4.4.3. Módulo Control de Accesos.....	106
4.4.4. Principal.....	107
4.4.5. Web Services.....	107
4.5. Diagramas de clases.....	107
4.5.1. Módulo Consola.....	107
4.5.2. Módulo GPS.....	111
4.5.3. Módulo Control de Accesos.....	114
4.5.4. Principal.....	118
4.5.5. Web Service.....	118
5. Diseño de la interfaz.....	120
5.1. Principal.....	120
5.1.1. Pantalla de login.....	121
5.1.2. Pantalla launcher.....	122
5.1.3. Pantalla falsa.....	123

5.2. Módulo Consola.....	123
5.2.1. Pantalla principal.....	124
5.2.2. Pestaña manual.....	125
5.2.3. Pestaña favoritos.....	126
5.2.4. Formulario.....	127
5.2.5. Terminal SSH.....	128
5.2.6. Pantalla comandos ejecutados.....	129
5.2.7. Pantalla comandos favoritos.....	130
5.3. Módulo GPS.....	131
5.3.1. Pantalla principal.....	131
5.4. Módulo Control de Accesos.....	132
5.4.1. Filtro.....	132
5.4.2. Formulario.....	133
5.4.3. Main.....	134
5.4.4. Pestaña favoritos.....	135
5.4.4.1. Lista servidores.....	136
5.4.4.5. Pestaña notificaciones.....	137
6. Diseño procedimental.....	137
6.1. Principal.....	138
6.1.1. Login.....	138
6.1.2. Main.....	139
6.2. Módulo SSH.....	140
6.2.1. Guardar nuevo servidor.....	140
6.2.2. Conectar a servidor.....	140
6.2.3. Enviar comando.....	142
6.2.4. Guardar script.....	143
6.2.5. Ejecutar script guardado.....	144
6.3. Módulo GPS.....	145
6.3.1. Iniciar desde sms.....	145
6.3.2. Guardar localizaciones.....	146
6.3.3. Enviar correo electrónico.....	147
6.3.4. Enviar sms.....	148
6.3.5. Autorranque al encender.....	149
6.4. Módulo Consola de Acessos.....	149
6.4.1. Añadir servidor SSH.....	149
6.4.2. Monitorizar servidor SSH.....	149
6.4.3. Descargar notificaciones nuevas y guardadas.....	150
6.4.4. Filtrar notificaciones.....	152
6.4.5. Eliminar filtro.....	154
6.4.6. Borrar notifcaciones.....	154
6.5. Web Service.....	156
6.5.1. Borrar notificación.....	156
6.5.2. Hay notificaciones.....	156
6.5.3. Obtener notificaciones.....	156
6.5.4. ScanLog.....	157
7. Referencias cruzadas a los requisitos.....	157
8. Pruebas.....	158

8.1. Pruebas de stress.....	158
8.2. Pruebas de integración.....	159
8.3. Tablas de pruebas.....	159
8.3.1. Principal.....	159
8.3.2. Módulo SSH.....	160
8.3.2.1. AccesoDatos.....	160
8.3.2.2. Vista.....	161
8.3.3. Módulo GPS.....	161
8.3.3.1. AccesoDatos.....	161
8.3.3.2. Vista.....	162
8.3.4. Módulo Control de Accesos.....	163
8.3.4.1. Vista.....	163
8.3.5. Web Service.....	163
9. Entorno tecnológico del sistema.....	163
9.1. Restricciones técnicas.....	164
9.2. Comunicación dentro del sistema.....	165

## Índice de ilustraciones

Ilustración 21: Fichero xml.....	94
Ilustración 22: Cliente/Servidor.....	99
Ilustración 23: Capas.....	100
Ilustración 24: Singleton Conexion.....	101
Ilustración 25: Fachada.....	102
Ilustración 26: Diagrama de paquetes.....	103
Ilustración 27: Paquetes.....	104
Ilustración 28: modulossal accesodatos.....	108
Ilustración 29: modulossal modelo.....	109
Ilustración 30: modulossal logica.....	110
Ilustración 31: modulologps accesodatos.....	111
Ilustración 32: modulologps modelo.....	112
Ilustración 33: modulologps logica.....	113
Ilustración 34: modulocontrol accesodatos.....	114
Ilustración 35: modulocontrol modelo.....	115
Ilustración 36: modulocontrol logica1.....	116
Ilustración 37: modulocontrol logica2.....	117
Ilustración 38: Principal.....	118
Ilustración 39: Web Service analizador.....	119
Ilustración 40: Web Service notificador.....	119
Ilustración 41: Principal login.....	121
Ilustración 42: Principal launcher.....	122
Ilustración 43: Principal falsa.....	123
Ilustración 44: ssh principal.....	124
Ilustración 45: ssh pestaña manual.....	125
Ilustración 46: ssh servidores favoritos.....	126
Ilustración 47: ssh formulario.....	127
Ilustración 48: ssh consola.....	128
Ilustración 49: ssh comandos ejecutados.....	129
Ilustración 50: ssh comandos favoritos.....	130
Ilustración 51: gps pantalla principal.....	131
Ilustración 52: control filtro.....	132
Ilustración 53: control formulario.....	133
Ilustración 54: control main.....	134
Ilustración 55: control pestaña.....	135
Ilustración 56: control lista servidores.....	136
Ilustración 57: control pestaña notificaciones.....	137
Ilustración 58: Secuencia principal login.....	138
Ilustración 59: principal secuencia main.....	139
Ilustración 60: ssh nuevo servidor.....	140
Ilustración 61: ssh conectar.....	141
Ilustración 62: ssh enviar comando.....	142
Ilustración 63: ssh guardar script.....	143
Ilustración 64: ssh ejecutar guardado.....	144

Ilustración 65: gps inicia desde sms.....	145
Ilustración 66: gps guardar localización.....	146
Ilustración 67: gps enviar correo electrónico.....	147
Ilustración 68: gps enviar sms.....	148
Ilustración 69: gps autoarranque.....	149
Ilustración 70: control monitorizar.....	150
Ilustración 71: control descargar.....	151
Ilustración 72: control filtrar.....	153
Ilustración 73: control quitar filtro.....	154
Ilustración 74: control borrar.....	155
Ilustración 75: Borrar notificación.....	156
Ilustración 76: Hay notificaciones.....	156
Ilustración 77: Obtener notificaciones.....	156
Ilustración 78: Scan Log.....	157
Ilustración 79: Diagrama componentes.....	164
Ilustración 80: Diagrama componentes completo.....	165

## Índice de tablas

Tabla 32: Notificación.....	96
Tabla 33: Dispositivo.....	96
Tabla 34: Notificación-Dispositivo.....	96
Tabla 35: Servidores.....	97
Tabla 36: Scripts.....	97
Tabla 37: Comandos.....	97
Tabla 38: Coordenada.....	98
Tabla 39: Servidores.....	98
Tabla 40: Email.....	98
Tabla 41: Referencia a los requisitos.....	158
Tabla 42: Pruebas principal.....	159
Tabla 43: Pruebas ssh accesodatos.....	160
Tabla 44: Pruebas ssh vista.....	161
Tabla 45: Pruebas gps vista.....	162
Tabla 46: Pruebas accesodatos vista.....	163
Tabla 47: Pruebas webservice.....	163

## LISTA DE CAMBIOS

Versión	Fecha	Descripción
1.0	16/03/12	Primera versión: Introducción.
2.0	30/03/12	Segunda versión: Diseño de datos.
3.0	22/04/12	Tercera versión : Añadidos todos los diagramas.
4.0	24/06/12	Cuarta versión.: Corrección de fallos.

## 1 INTRODUCCIÓN

Este documento pretende ser la entrada en el dominio de la solución de problemas, es decir, el inicio del desarrollo del sistema software.

Se trata de definir la arquitectura, los datos, la interfaz o los detalles procedimentales de la aplicación software.

El proceso de diseño de una aplicación es fundamental para garantizar la calidad de un producto software, ya que es aquí donde se puede establecer la diferencia entre un buen sistema software o un mal sistema. Han de tenerse en cuenta los siguientes factores:

- ¿Como se va a desplegar nuestro sistema?
- ¿Como van a utilizar los usuarios nuestro sistema?
- ¿Que cambios en el sistema pueden producir un problema?

Además, teniendo en cuenta que nuestro sistema se desarrolla en torno a módulos, cada una de las cuestiones anteriores se tratará de forma individual para cada módulo, y si fuera necesario, un resumen global con todo el sistema ensamblado.

## 2 ÁMBITO DE SOFTWARE

El objetivo más importante de esta aplicación es la de controlar y monitorizar de la forma más completa posible los distintos equipos servidores que una empresa pueda tener a su cargo, y de forma más concreta, ayudar al administrador en el mantenimiento de dichos equipos.

Toda este control se realizará desde un dispositivo móvil, por lo que todo el sistema estará condicionado a las restricciones inherentes del propio dispositivo.(pantallas pequeñas y poca potencia para procesar datos).

### 1. Módulo Consola

El objetivo mas importante de este módulo será el de poder controlar el servidor de forma remota, simplemente enviando comandos y ejecutando diferentes scripts.

### 2. Módulo GPS

El objetivo mas importante de este módulo es el de localizar y hacer el seguimiento de un dispositivo android.

### 3. Módulo Control de Accesos

Su objetivo principal es la monitorización en tiempo real del control de accesos de usuarios a un servidor, tanto si el intento ha tenido éxito como si no. Además de monitorizar los mensajes salientes por si ocurren situaciones extrañas como envíos masivos.

#### 4. Web Service

El objetivo principal del Web Service es el de filtrar el syslog en busca de eventos que tratar y guardarlos en una base de datos, y también el responder a las llamadas que se le hagan desde el exterior y enviarle las notificaciones pertinentes.

### 3 DISEÑO DE DATOS

El objetivo del diseño de datos es seleccionar representaciones lógicas de los objetos de datos (estructuras de datos) identificados en la fase de análisis de requisitos.

Dependiendo del tipo de aplicación que se esté desarrollando se pueden tener diferentes aproximaciones en cuanto a los datos, pudiéndose distinguir en general:

- Objetos de datos
- Estructuras de datos
- Estructuras de archivo
- Bases de datos

#### 3.1 Objetos de datos y estructuras de datos

Se trata de definir un diccionario de datos para representar las relaciones entre los objetos de datos.

##### 3.1.1 Módulo Consola SSH

###### Script:

idScript: (int) identificador del Script, único.

Nombre: (String) nombre del script.

Cantidad: (int) cantidad de comandos individuales que contiene el script.

Comandos: (List<String>) lista de comandos que forman el script.

###### Servidor:

id:(int) identificador del servidor, único.

IP: (String) ip del servidor.

Puerto: (int) puerto por el que se conectará a ese servidor, normalmente puerto 22.

Usuario: (String) usuario con el que se conectará al servidor.

Contraseña: (String) contraseña del usuario con el que se conecta.

Descripcion: (String) pequeña descripción del servidor.

**SingletonConexion:**

session: (Session) sesión creada con un servidor concreto.

Jsch: (JSch) toda la información de la sesión encapsulada.

**3.1.2 Módulo Control de Accesos****Notificacion:**

id:(int) identificador de la notificación en la base de datos remota.

Serv: (Servidor) servidor asociado a esa notificación.

Texto: (String) texto que causó la notificación.

Tipo: (int) tipo de notificación, 0 ssh, 1 email

Urgencia: (int) urgencia, 0 grave, 1 medio, 2 información.

Fecha: (Date) fecha en la que se creó la notificación, en GMT.

**Servidor:**

id: (int) identificador del servidor.

Ip: (String) ip del servidor.

Puerto: (int) puerto al que conectarse para recibir notificaciones.

Inicio: (boolean) indica si se autoarranca o no al iniciar el teléfono.

Color: (int) color que se asigna para identificarlo de forma más rápida.

Descripcion: (String) pequeña descripción del servidor..

**SingletonEmail:**

hosts:(List<Servidor>) lista con todos los servidores a consultar.

**SingletonServicios:**

hosts: (List<Servidor>) lista con todos los servidores a consultar.

**3.2 Estructuras de archivos y bases de datos**

Se define por una parte el modelo lógico (modelo relacional, normalización..) y por otro el modelo físico (creación de índices, propiedades específicas de atributos...). Siempre teniendo en cuenta la limitación que ofrece SQLite como base de datos.

La estructura de archivos xml que se utilizan para guardar las preferencias de la aplicación son los mismos que usa Android ya que es el propio sistema el que se encarga de crearlos y manejarlos de forma transparente para el programador.

La estructura de un fichero xml de configuración en Android tiene la forma:

```
<map>
    <[boolean,long,int] value=""[valor]" name=""[nombre campo]""/>
    <string name=""[nombre campo]"">[valor]</string>
</map>
```

Como ejemplo:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <map> .
    <string name="pref_camera_recordlocation_key">off</string>
    <boolean name="search_shutter_enabled" value="true"/>
    <string name="timer_duration">10</string>
    <boolean name="power_shutter_enabled" value="true"/>
    <boolean name="vol_up_shutter_enabled" value="false"/>
    <boolean name="vol_zoom_enabled" value="true"/>
    <string name="pref_camera_jpegquality_key">superfine</string>
    <int name="pref_version_key" value="4"/>
    <boolean name="pre_focus_enabled" value="true"/>
</map>
```

*Ilustración 21: Fichero xml*

En cuanto a la estructura con la que se reciben los mensajes de log de la aplicación syslog es la siguiente:

*Fecha + equipo + servicio que alerta + [Puerto] + Mensaje*

Un ejemplo de un mensaje syslog es:

Feb 23 09:25:45 server sshd[12345]: Failed password for root from 123.45.67.89 port 12345 ssh2.

En cuanto a bases de datos usamos SQLite, con las consiguientes restricciones que conlleva, como la ausencia de algunos tipos de datos.

Los tipos de datos disponibles en SQLite Versión 3 son:

- NULL
- INTEGER: número entero, de 1,2,3,4,6,o 8 bytes dependiendo del tamaño.
- REAL: número en coma flotante.
- TEXT: texto guardado en formato UTF-8, UTF-16BE o UTF-16LE
- BLOB: conjunto de bytes guardado exactamente como se introdujo.

### 3.2.1 Web Service

El Web Service trabaja con una base de datos relacional completa, por lo que no sufre de las carencias de la base de datos SQLite.

Por otra parte el Web Service consta de dos partes a su vez, el Web Service que responde a las llamadas y un demonio que se encarga de filtrar los mensajes de log, cada uno de ellos tiene un archivo de configuración.

Fichero de configuración del Web Service:

```
#Servidor de base de datos
dataBaseServer=localhost

# Nombre de la base de datos
dataBaseName=android_server_status

# Usuario de la base de datos
dataBaseUser=jonatan

# Contraseña del usuario
dataBasePassword=prueba
```

Fichero de configuración del demonio:

```
# Servidor de base de datos
dataBaseServer=localhost

# Nombre de la base de datos
dataBaseName=android_server_status

# Usuario de la base de datos
dataBaseUser=jonatan

# Contraseña del usuario
dataBasePassword=prueba

# Puerto de escucha de mensajes syslog
portIn=5150

# Tiempo en milisegundos de comprobación de nuevos mensajes
tiempoComprobacion=1000

# Número de emails enviados por minutos que se considera spam
numeroMails=6

# Número de intentos de conexión al correo por minuto que se considera ataque
numeroAtaque=6
```

### 3.2.1.1 Modelo lógico

Notificación(idNotificación “PK”, TipoMensaje, Urgencia, Fecha, Mensaje)

Dispositivo(idDispositivo “PK”, Dispositivo)

DispositivoNotificación(idNotificación “FK”, idDispositivo “FK”, Descargada, PRIMARYKEY(idNotificación,idDispositivo))

### 3.2.1.2 Modelo físico

Campo	Tipo	Extra
idNotificación	INTEGER	Primary key, autoincrement
TipoMensaje	INTEGER	NOT NULL
Urgencia	INTEGER	NOT NULL
Fecha	TIMESTAMP	Default: CURRENT TIME STAMP
Mensaje	LONGTEXT	

Tabla 32: Notificación

Campo	Tipo	Extra
idDispositivo	INTEGER	Primary key, autoincrement
Dispositivo	CHAR(16)	NOT NULL

Tabla 33: Dispositivo

Campo	Tipo	Extra
idNotificación	INTEGER	Foreign key (Notificacion), PK
idDispositivo	INTEGER	Foreign key (Dispositivo), PK
Descargada	tinyint	Default: 0

Tabla 34: Notificación-Dispositivo

### 3.2.2 Módulo Consola

#### 3.2.2.1 Modelo lógico

servidores(id “PK”, host, port, user, pass, descripcion)

scripts(idScript “PK”, nombre, cantidad)

comandos(idComando “PK”, idScript ”FK scripts”, comando)

### 3.2.2.2 Modelo físico

Servidores:

Campo	Tipo	Extra
id	INTEGER	Primary key, autoincrement
host	TEXT	
port	TEXT	
user	TEXT	
pass	TEXT	
descripcion	TEXT	

Tabla 35: Servidores

Scripts:

Campo	Tipo	Extra
idScript	INTEGER	Primary key, autoincrement
nombre	TEXT	
cantidad	TEXT	

Tabla 36: Scripts

Comandos:

Campo	Tipo	Extra
idComando	INTEGER	Primary key, autoincrement
idScript	INTEGER	Foreign key references Scripts, ON UPDATE/DELETE CASCADE
comando	TEXT	

Tabla 37: Comandos

### 3.2.3 Módulo GPS

#### 3.2.3.1 Modelo lógico

coordenadas(idCoordenada “PK”, longitud, latitud, fecha)

### 3.2.3.2 Modelo físico

Campo	Tipo	Extra
idCoordenada	INTEGER	Primary key, autoincrement
longitud	REAL	
latitud	REAL	
fecha	INTEGER	Actua como Long aunque se declare como INTEGER

Tabla 38: Coordenada

### 3.2.4 Módulo Control de Accesos

#### 3.2.4.1 Modelo lógico

servidores(id “PK”, host, inicio, color, descripcion, puerto);  
email(id “PK”, host, inicio, color, descripcion, puerto);

#### 3.2.4.2 Modelo físico

Campo	Tipo	Extra
id	INTEGER	Primary key, autoincrement
host	TEXT	
inicio	INTEGER	0 false, 1 true
color	INTEGER	
descripcion	TEXT	
puerto	INTEGER	

Tabla 39: Servidores

Campo	Tipo	Extra
id	INTEGER	Primary key, autoincrement
host	TEXT	
inicio	INTEGER	0 false, 1 true
color	INTEGER	
descripcion	TEXT	
puerto	INTEGER	

Tabla 40: Email

## 4 DISEÑO ARQUITECTÓNICO

El diseño arquitectónico tiene como objetivo desarrollar la estructura modular que representa las relaciones entre los módulo, combinando la estructura del programa con la de los datos a través de interfaces que permiten el flujo de éstos.

La arquitectura de la aplicación depende completamente del problema concreto a resolver. La mayoría de las veces se utilizan varios estilos arquitectónicos para la solución.

### 4.1 Estilos arquitectónicos utilizados

#### 4.1.1 Cliente-Servidor

El estilo Cliente/Servidor divide la aplicación en dos partes: una de ellas, el cliente, realiza peticiones a la otra, el servidor, que procesa la petición y envía una respuesta al cliente.

##### Características y ventajas:

Se utiliza en sistemas distribuidos.

El cliente puede hacer peticiones a varios servidores.

El servidor nunca pregunta al cliente, por lo que no es necesario que sepa cuantos clientes tiene ni la forma en la que actúan.

El usuario solo necesita interactuar con el cliente.

Todos los cálculos los realiza el servidor, por lo que se libera carga del cliente.

Todos los usuarios trabajarán sobre los mismo datos, ya que hacen peticiones a los mismos servidores.

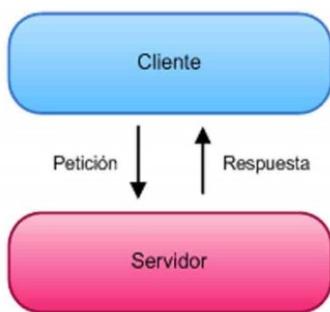


Ilustración 22: Cliente/Servidor

En este caso, se utiliza la arquitectura Cliente/Servidor en el módulo de Control de Accesos, el dispositivo móvil se encarga de comunicarse con el servidor cada cierto tiempo para conocer el estado del mismo. En el lado del servidor se ha implementado un Web Service que es el que se encarga de comparar todos los ficheros y de dar la respuesta a las peticiones del dispositivo móvil.

Con esto conseguimos liberar de carga de procesamiento al dispositivo móvil, y toda esta tarea recae sobre el servidor que es seguro tendrá mucha más potencia de cálculo.

#### 4.1.2 Arquitectura en capas

Este tipo de arquitectura distribuye jerárquicamente los roles y las responsabilidades dentro de la aplicación. De esta forma se proporciona una división de los problemas a resolver.

**Ventajas:** Abstracción, aislamiento, rendimiento y testeabilidad.

En nuestra aplicación, y debido a que el desarrollo se hace sobre dispositivos móviles, con sus propias peculiaridades, nos hemos decantado por separar la parte de acceso a datos permanentes, clases entity, y clases GUI + funcionalidad.

En Android la parte gráfica y de funcionalidad está íntimamente ligada, un cambio en la interfaz suele obligar a cambiar también parte de código destinado a dar funcionalidad, por lo que hemos decidido no separar estas dos partes. Aunque a pesar de todo esto, utilizamos distintos mecanismos (clases internas, métodos privados,...) para separar tanto como podamos los dos puntos de vista, GUI y lógica del programa, para intentar minimizar el acoplamiento.

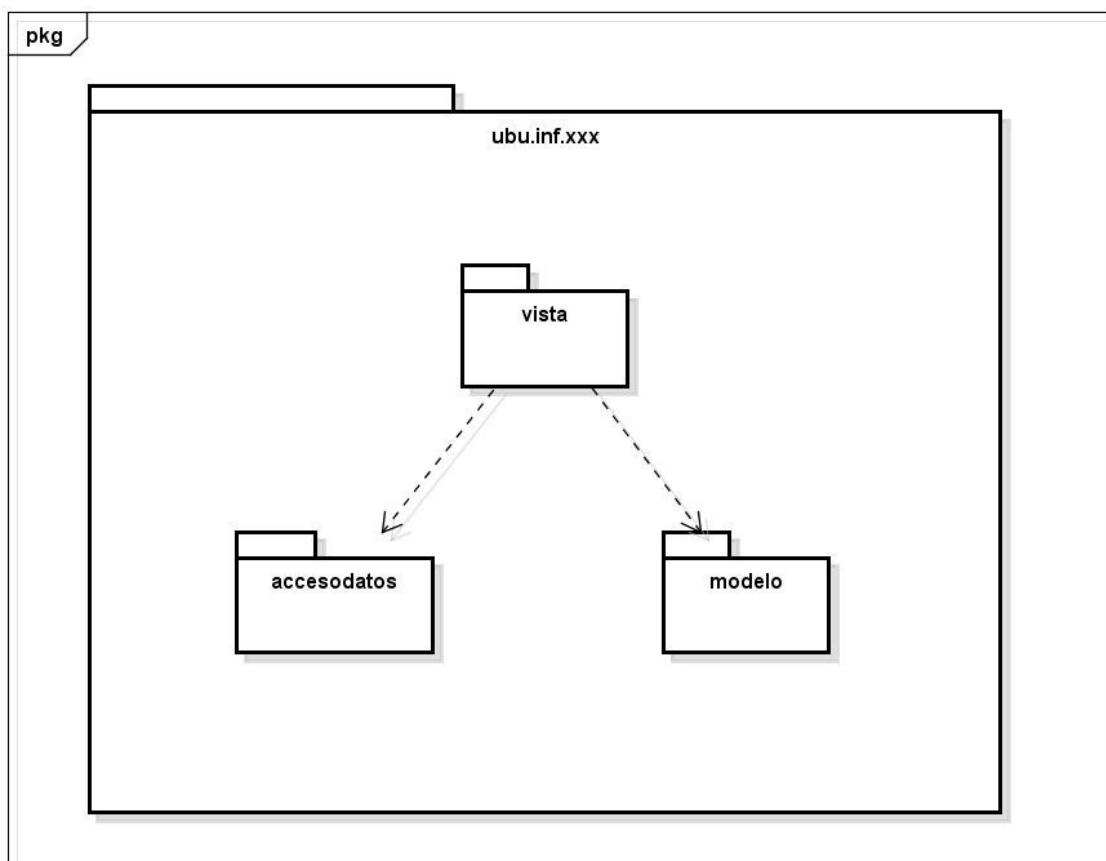


Ilustración 23: Capas

## 4.2 Patrones de diseño utilizados

Ahora se describirán algunos de los patrones de diseño utilizados, y uno de los ejemplos donde se han usado (pueden haberse usado en más de un lugar).

### 4.2.1 Singleton

Asegura que solo se crea una única instancia de una clase. Todos los objetos que utilicen una instancia de la clase será la misma.

El constructor de la clase se declara privado, la clase tiene una variable estática que referencia a la única instancia de la clase, se crea un método público y estático que devuelve el valor de esa referencia.

Se ha utilizado por ejemplo, en el moduloConsola, en la clase “SingletonConexion”.

Esta clase se comporta de forma especial, no tiene funcionalidad, solo contiene unas variables para guardar la referencia de otros objetos. Se ha usado para pasar información compleja entre distintas actividades, una de las actividades guarda mediante un método setXXX(); la información a guardar, cuando otra clase quiere leer la información almacenada solo tiene que usar el método getXXX(); apropiado a la situación.

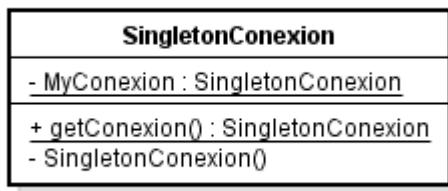


Ilustración 24: Singleton Conexion

### 4.2.2 Fachada

El patrón de diseño Fachada simplifica el acceso a un conjunto de objetos relacionados proporcionando un único objeto con el que los objetos externos puedan interactuar con el subsistema.

En nuestro caso hemos utilizado el patrón Fachada para aislar a la aplicación el uso de la base de datos, sin tener que usar comandos sql ya que es la propia fachada la que contiene toda esa información. Igualmente, al usar una fachada, podríamos hacer que los datos se guardaran en otros sitios, tales como ficheros xml, o bases de datos remotas sin tener que cambiar la lógica del programa, solo modificando la fachada.

La fachada contiene los típicos métodos asociados a una base de datos, como son:

- load, obtiene todas las filas de la base de datos
- insert, inserta un nuevo dato
- edit, edita una fila ya existente

- delete, borra una fila por su id

Como ejemplo ponemos la fachada correspondiente al ModuloGPS:

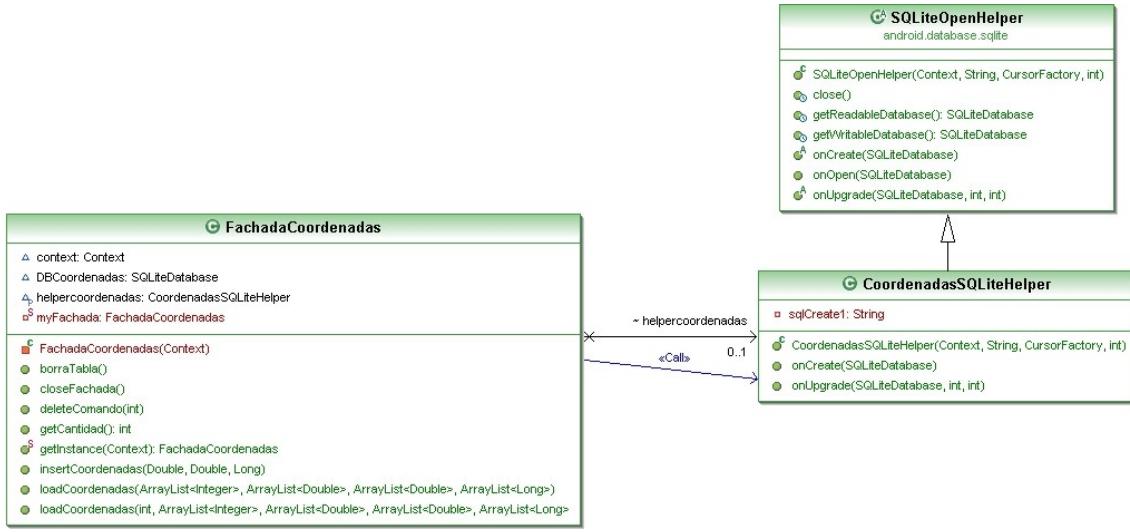


Ilustración 25: Fachada

### 4.3 Diseño general

La aplicación, como bien se ha comentado a lo largo de esta misma memoria, consta de diferentes módulos que cuentan con un alto grado de independencia entre ellos, gracias a esta característica se hace sencillo el poder añadir más funcionalidad a la aplicación solamente con añadir más módulos.

En el diseño general, una aplicación principal actúa de lanzador, y se encargará de iniciar el resto de módulos cuando el usuario quiera.

Además hay que tener claras las dos partes de la aplicación que existen.

Por un lado está el Web Service junto al analizador de logs, los cuales están desarrollados completamente en Java y se instalarán en un servidor Linux, por otro lado está el cliente, el cual ha sido desarrollado para trabajar bajo el sistema operativo Android, y que se encargará de mostrar los datos recibidos del Web Service.

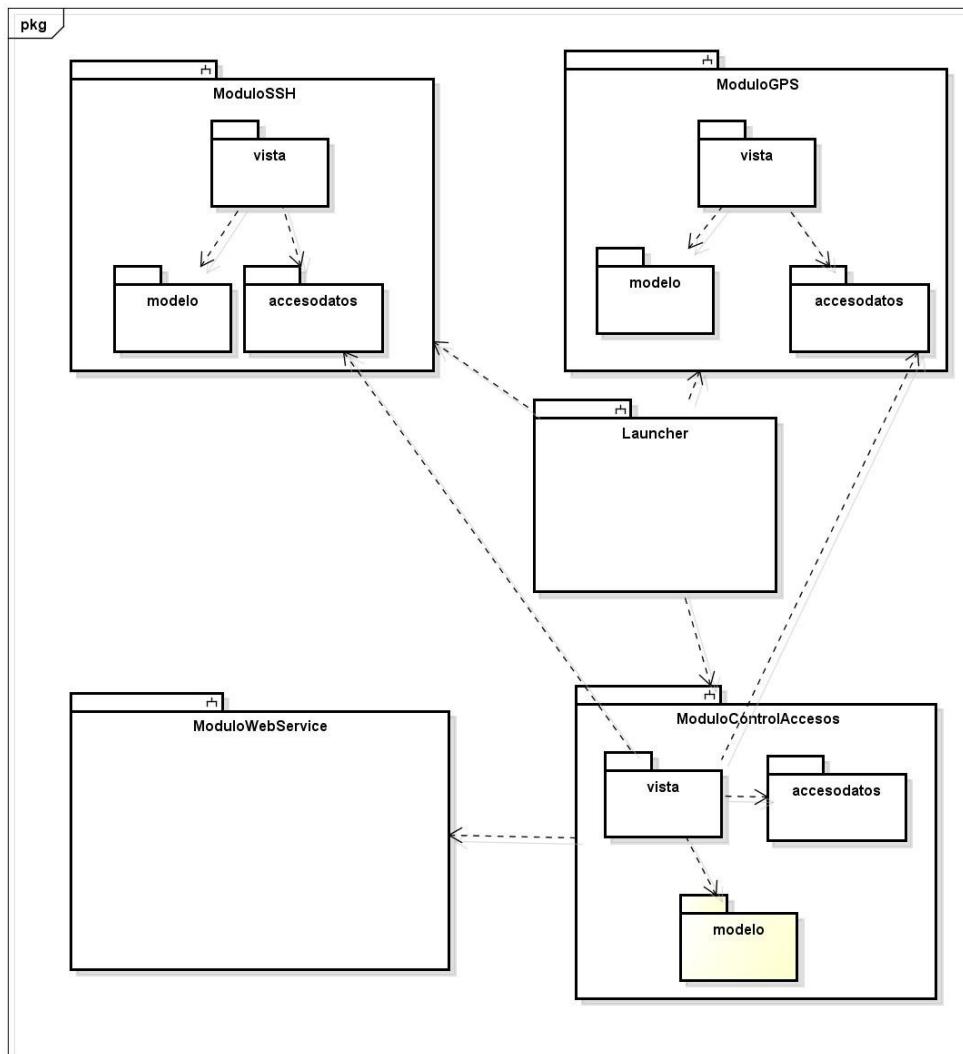


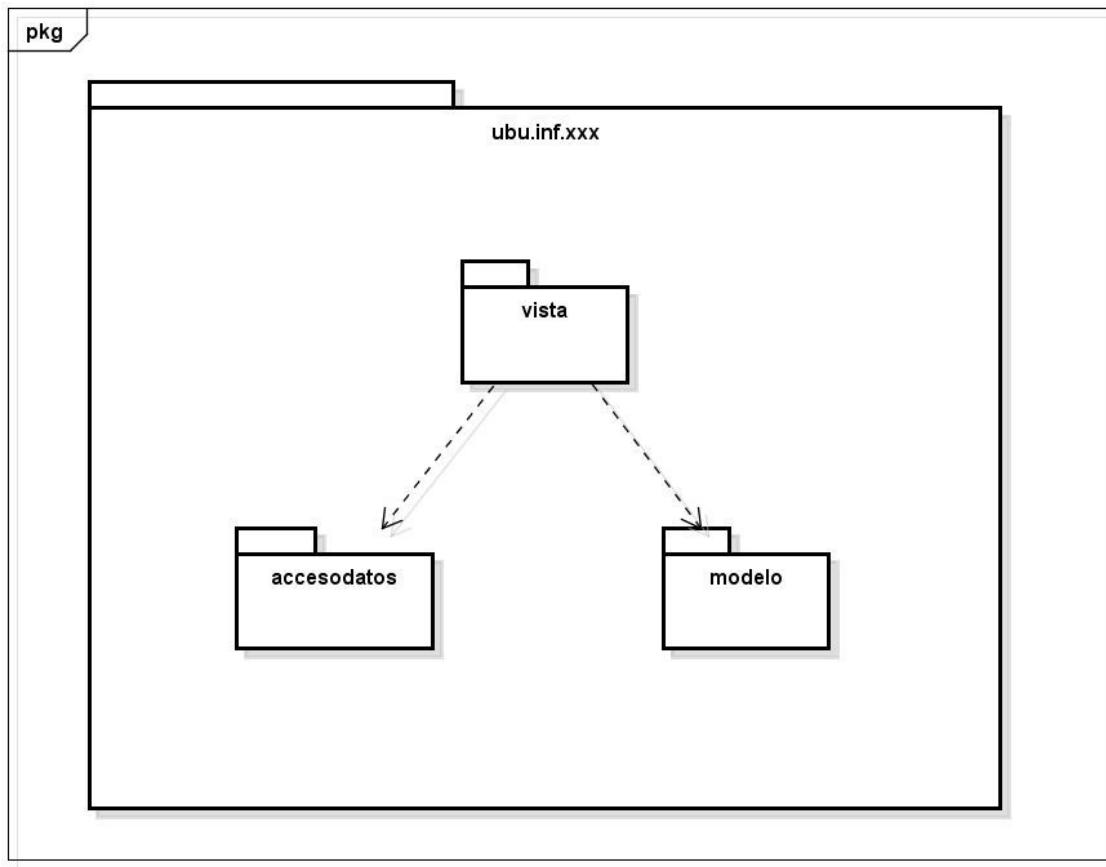
Ilustración 26: Diagrama de paquetes

#### 4.4 Descripción detallada

Esta es la parte más completa y con más información. Aquí se indicarán los distintos paquetes que componen cada módulo, las clases que contienen y la funcionalidad general de cada una de ellas.

Como en anteriores apartados, todo el texto estará estructurado en torno a los distintos módulos incluidos en la aplicación.

Tanto el Módulo Consola, como GPS y Control de Accesos tienen la misma estructura de paquetes, solo diferenciándose en las clases que contienen cada uno de ellos:



*Ilustración 27: Paquetes*

Además en los tres módulos existen 3 tipos de clases, las que solo realizan acciones y que son usadas por otras clases mediante sus métodos públicos, las pantallas gráficas (activity), y las clases que solo sirven para guardar información (entity).

Las clases internas son usadas para crear Listener, Adapters o Threads asíncronos.

#### 4.4.1 Módulo Consola

##### Accesodatos:

ComandosSQLiteHelper: clase que extiende de SQLiteHelper y sirve para crear las bases de datos de comandos.

FachadaComandos: clase que agrupa todos los métodos necesarios para manejar la base de datos de Comandos.

FachadaServidores: clase que agrupa todos los métodos necesario para manejar la base de datos de Servidores.

ServicioBorrado: servicio que borra todos los datos de la base de datos Comandos y Servidores.

ServidoresSQLiteHelper: clase que extiende de SqliteHelper y sirve para crear la

base de datos de Servidores.

**Logica:**

ComandosEjecutados: pantalla en la que se muestran los comandos ejecutados en la sesión actual para poder guardarlos como un script.

ComandosFavoritos: pantalla en la que se muestran los scripts favoritos guardados en la base de datos.

Consola: pantalla que emula a una terminal ssh.

Formulario: pantalla para introducir los datos de un nuevo Servidor, que posteriormente será guardado en la base de datos.

MainSSH: pantalla principal que contiene las dos pestañas.

PestanaMainFav: pestaña en la que se muestra una lista con todos los servidores guardados.

PestanaMainManual: pestaña para conectarse a un servidor de forma manual.

SUserInfo: clase de ayuda para Jsch.

**Modelo:**

Script: clase entity para guardar los datos de un script.

Servidor: clase entity para guardar los datos de un servidor.

SingletonConexión: clase que sirve de apoyo para guardar datos de forma temporal.

#### 4.4.2 Módulo GPS

**Accesodatos:**

CoordenadasSQLiteHelper: clase para crear la base de datos de Coordenadas.

FachadaCoordenadas: clase que agrupa los métodos necesarios para manejar la base de datos de Coordenadas.

**Logica:**

AutoArranque: clase que se ejecuta al encender el teléfono y arranca los servicios si es necesario.

LectorSMS: clase que lee los sms entrantes por si se manda alguna orden.

LocalizadorGPSActivity: pantalla principal de la aplicación.

Preferencias: pantalla con todas las preferencias y ajustes de la aplicación.

ServicioEnvioEmail: servicio que sirve para enviar Email en background.

ServicioEnvioSMS: servicio que sirve para enviar SMS en background.

ServicioGPS: servicio para obtener las coordenadas GPS y guardarlas en la base de datos de Coordenadas.

**Modelo:**

GmailSender: clase de ayuda para enviar emails desde una cuenta de google.

JSSEProvider: clase de ayuda a GmailSender.

#### 4.4.3 Módulo Control de Accesos

##### Accesos:

FachadaEmail: clase que agrupa todos los métodos para manejar la base de datos de servidores Email.

FachadaSSH: clase que agrupa todos los métodos para manejar la base de datos de servidores ssh.

ServicioBorrado: servicio para borrar todos los datos de las bases de datos de Email y Servidores.

ServidoresSQLiteHelper: clase para crear las bases de datos de Email y Servidores.

##### Logica:

AutoArranque: clase que se inicia al encender el teléfono por si es necesario arrancar el servicio ServicioAutoarranque.

Filtro: pantalla para establecer los criterios de filtrado para la lista de Notificaciones.

Formulario: pantalla para introducir los datos de un nuevo Servidor.

Main: pantalla que contiene las 3 pestañas.

PestanaMainEmail: pestaña con una lista de servidores a los que se puede controlar los emails.

PestanaMainSSH: pestaña con una lista de servidores a los que se puede controlar los accesos ssh.

PestanaMainNotificaciones: pestaña con una lista de todas las Notificaciones recibidas.

Preferencias: pantalla para modificar todas las preferencias y ajustes de la aplicación.

ServicioAutoarranque: servicio que comprueba las preferencias y arranca los servicios necesarios.

ServicioEmail: servicio que se encarga de consultar a los servidores para controlar los emails.

ServicioSSH: servicio que se encarga de consultar a los servidores para controlar los accesos ssh.

##### Modelo:

ComparatorFecha: clase que compara dos notificaciones por su campo fecha.

ComparatorID: clase que compara dos notificaciones por su campo id.

ComparatorTipo: clase que compara dos notificaciones por su campo Tipo.

ComparatorUrgencia: clase que compara dos notificaciones por su campo Urgencia.

Notificacion: clase entity para guardar todos los datos de una Notificación.

Servidor: clase entity para guardar todos los datos de un Servidor.

SingletonEmail: clase de ayuda para almacenar de forma temporal los servidores a los que hay que consultar los envíos de email.

SingletonSSH: clase de ayuda para almacenar de forma temporal los servidores a los que hay que consultar los accesos por ssh.

#### **4.4.4 Principal**

FalsaActividad: pantalla falsa que se muestra cuando se falla la contraseña.

Login: pantalla en la que introducir la contraseña de acceso.

Main: pantalla con enlaces a los tres módulos de la aplicación Android.

Preferencias: pantalla con la configuración de la aplicación.

#### **4.4.5 Web Services**

El Web Service se compone del Web Service propiamente dicho y del demonio que se encarga de analizar el log del sistema. En ambos casos se instalarán en un servidor Linux y son completamente independientes de la plataforma Android.

##### **Paquete notificador:**

Notificador: clase correspondiente al Web Service que responde las peticiones desde el exterior.

##### **Paquete scanlog:**

ScanLog: clase que implementa el demonio que se encarga de escanear el log del sistema.

### **4.5 Diagramas de clases**

En este apartado se incluirán los diagramas de clases junto con sus relaciones de todos los módulos de la aplicación. Si es necesario se incluirá alguna aclaración para comprender el funcionamiento de la aplicación.

#### **4.5.1 Módulo Consola**

Paquete accesodatos:

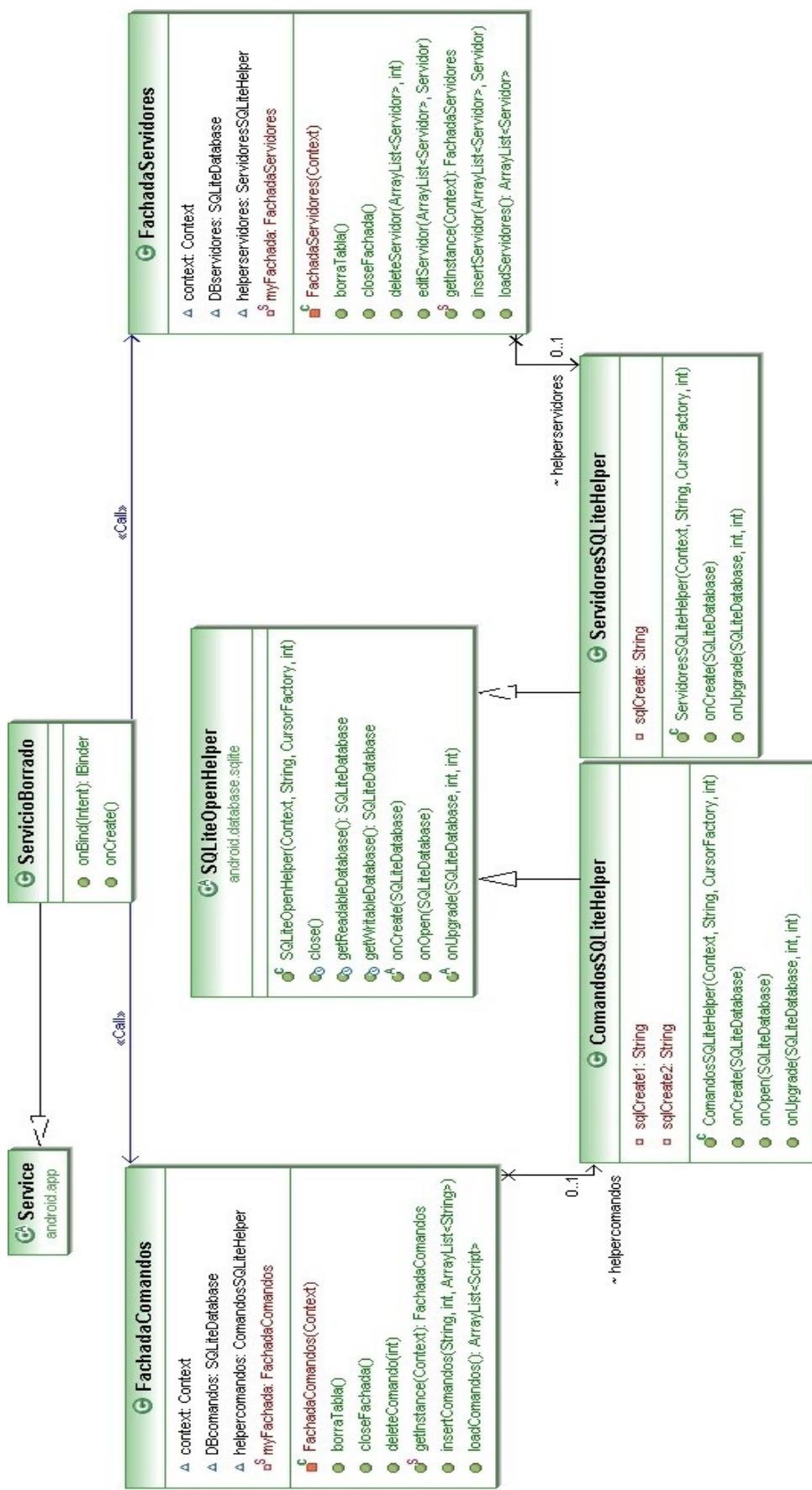


Ilustración 28: modulos sh accesados

Paquete modelo:

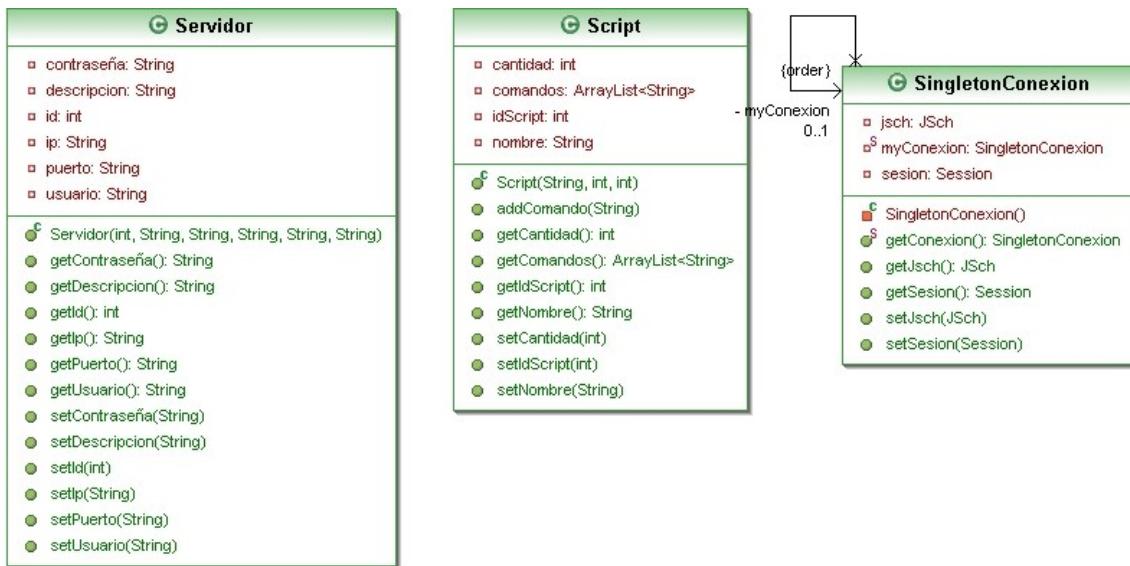
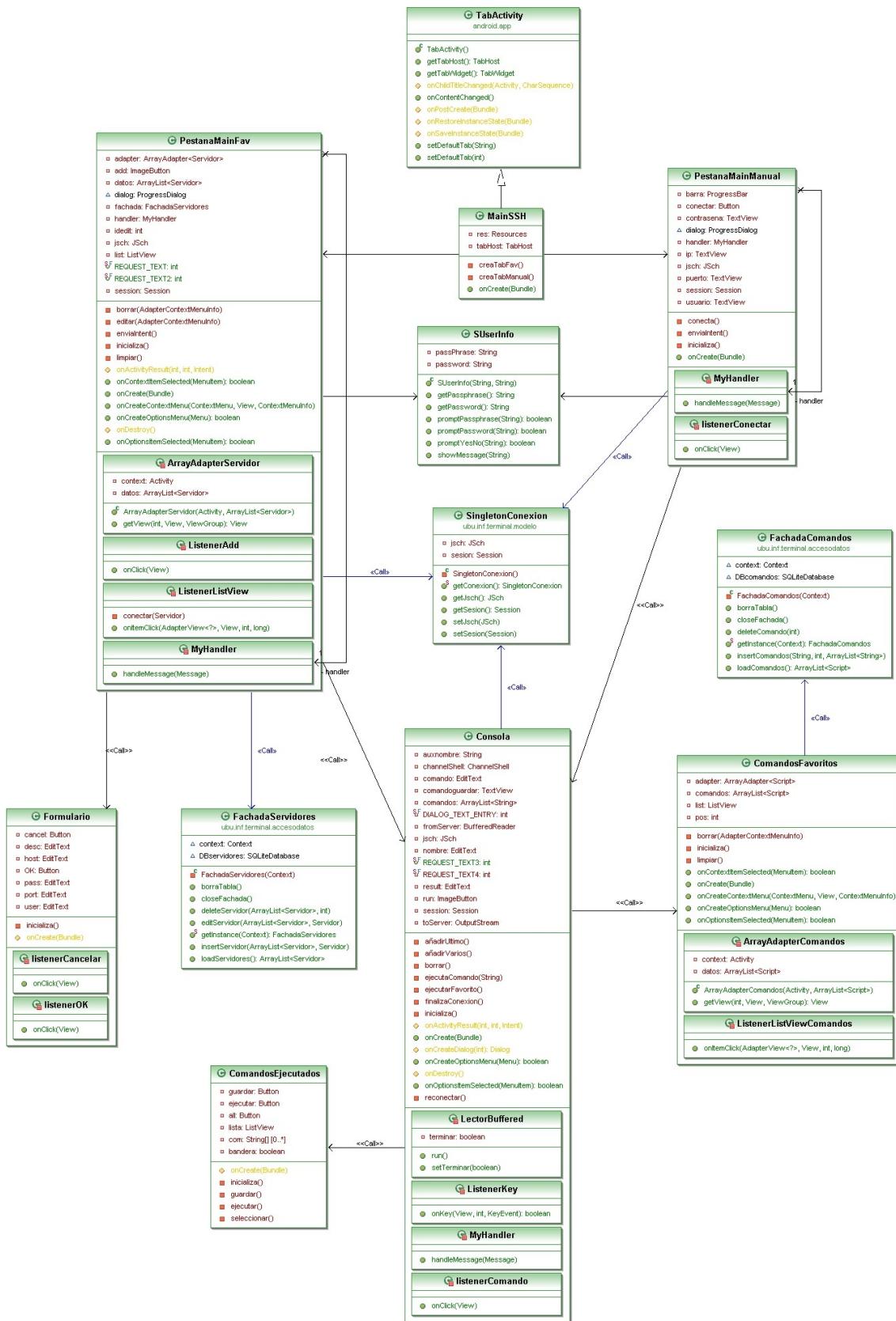


Ilustración 29: modulosssh modelo

Paquete logica:



*Ilustración 30: modulosh logica*

### 4.5.2 Módulo GPS

Paquete accesodatos:

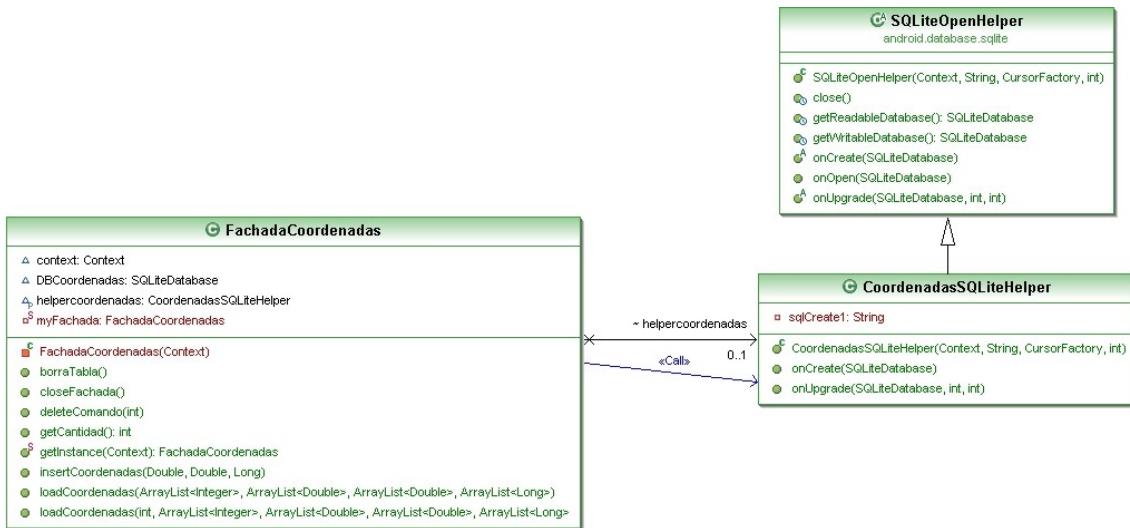


Ilustración 31: modulogps accesodatos

Paquete modelo:

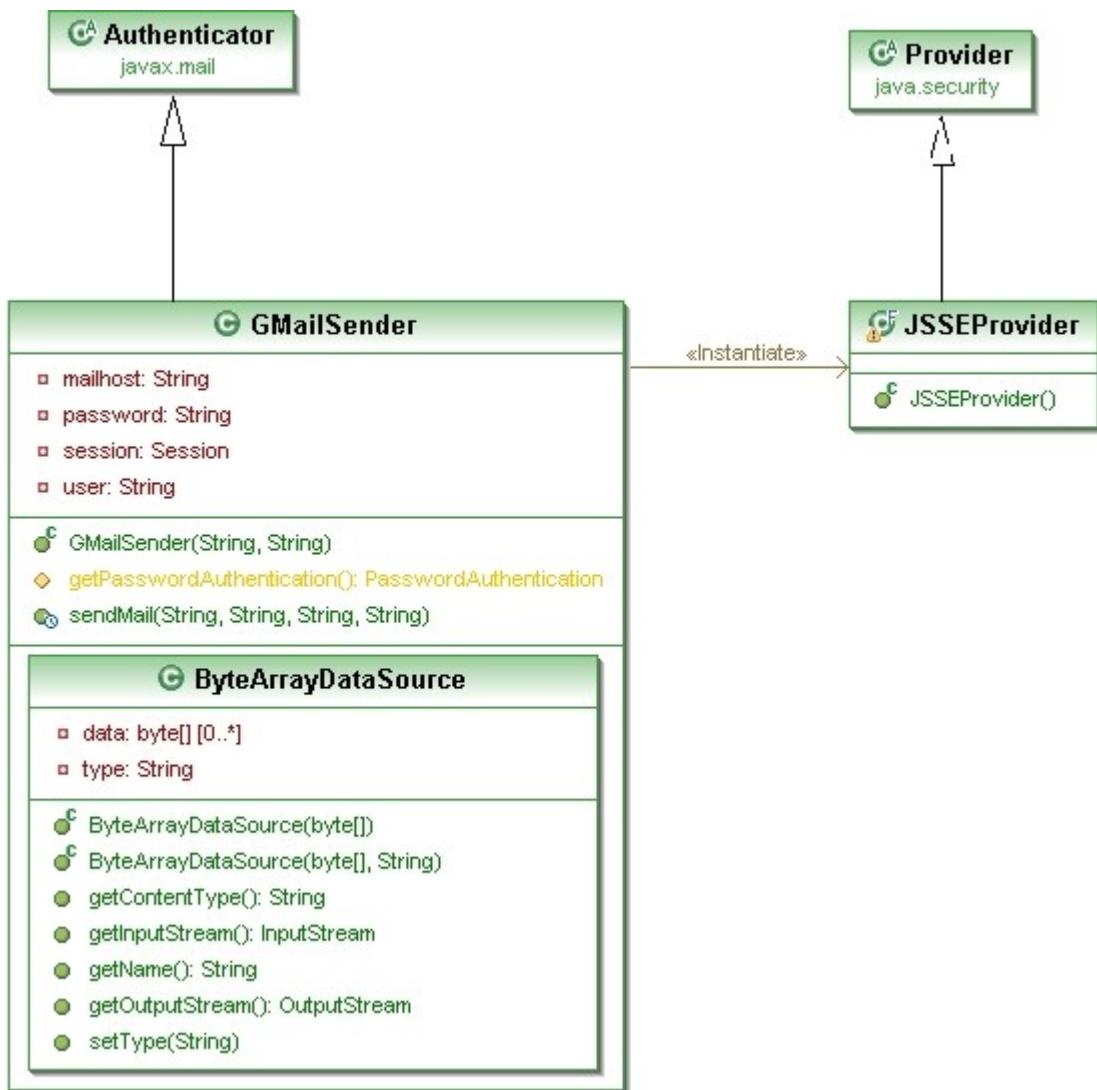


Ilustración 32: modulos del modelo

Paquete logica:

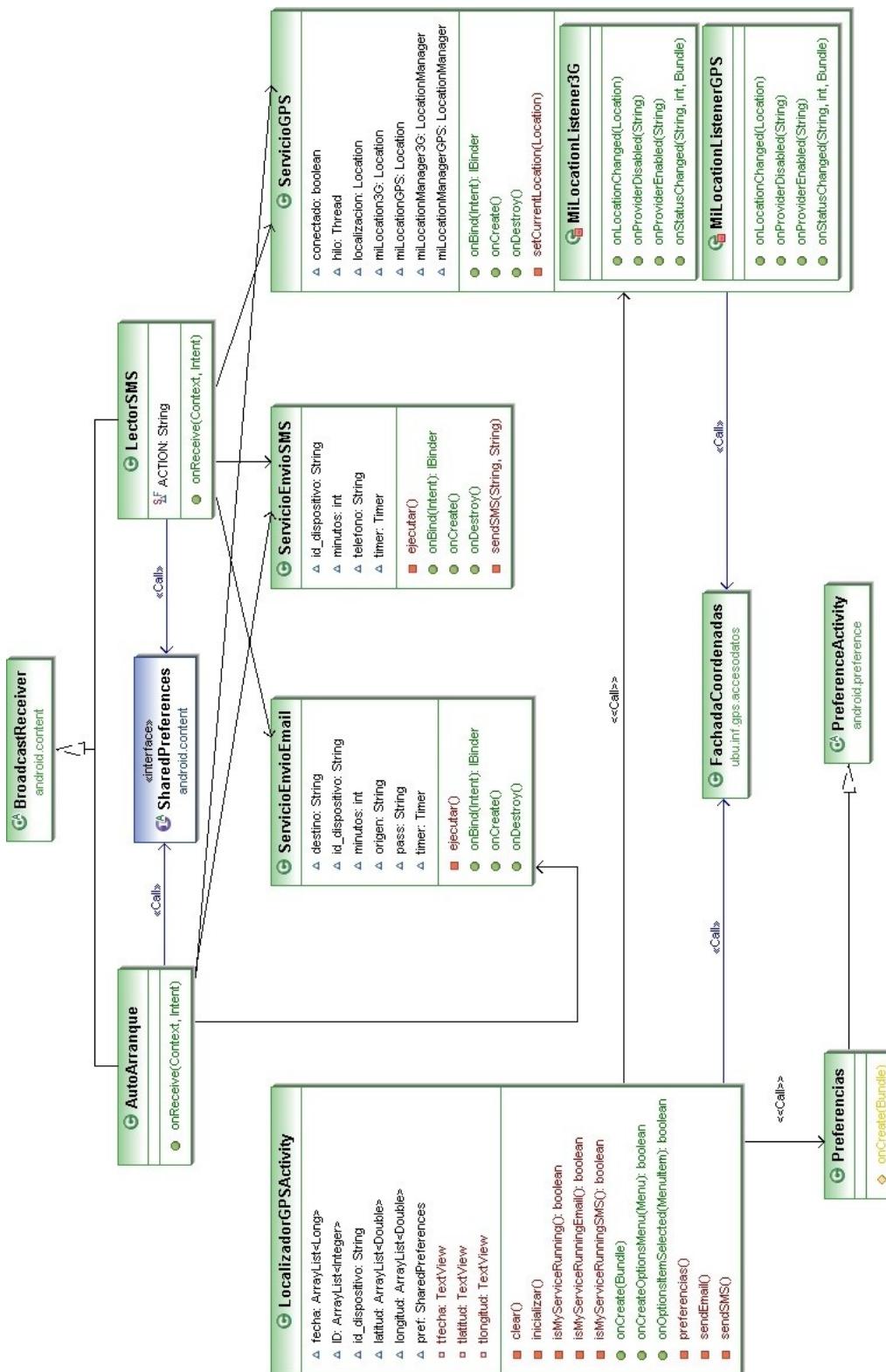


Ilustración 33: modulologps logica

### 4.5.3 Módulo Control de Acessos

Paquete accesodatos:

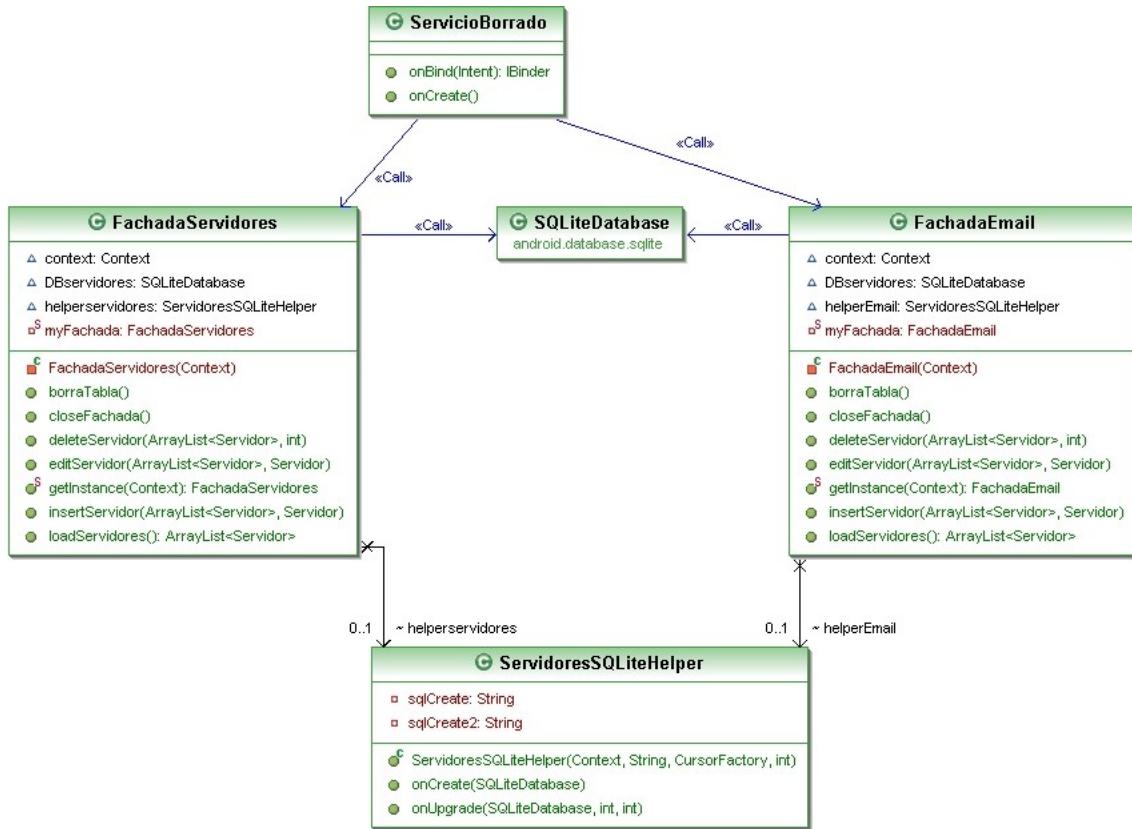


Ilustración 34: modulocontrol accesodatos

Paquete modelo:

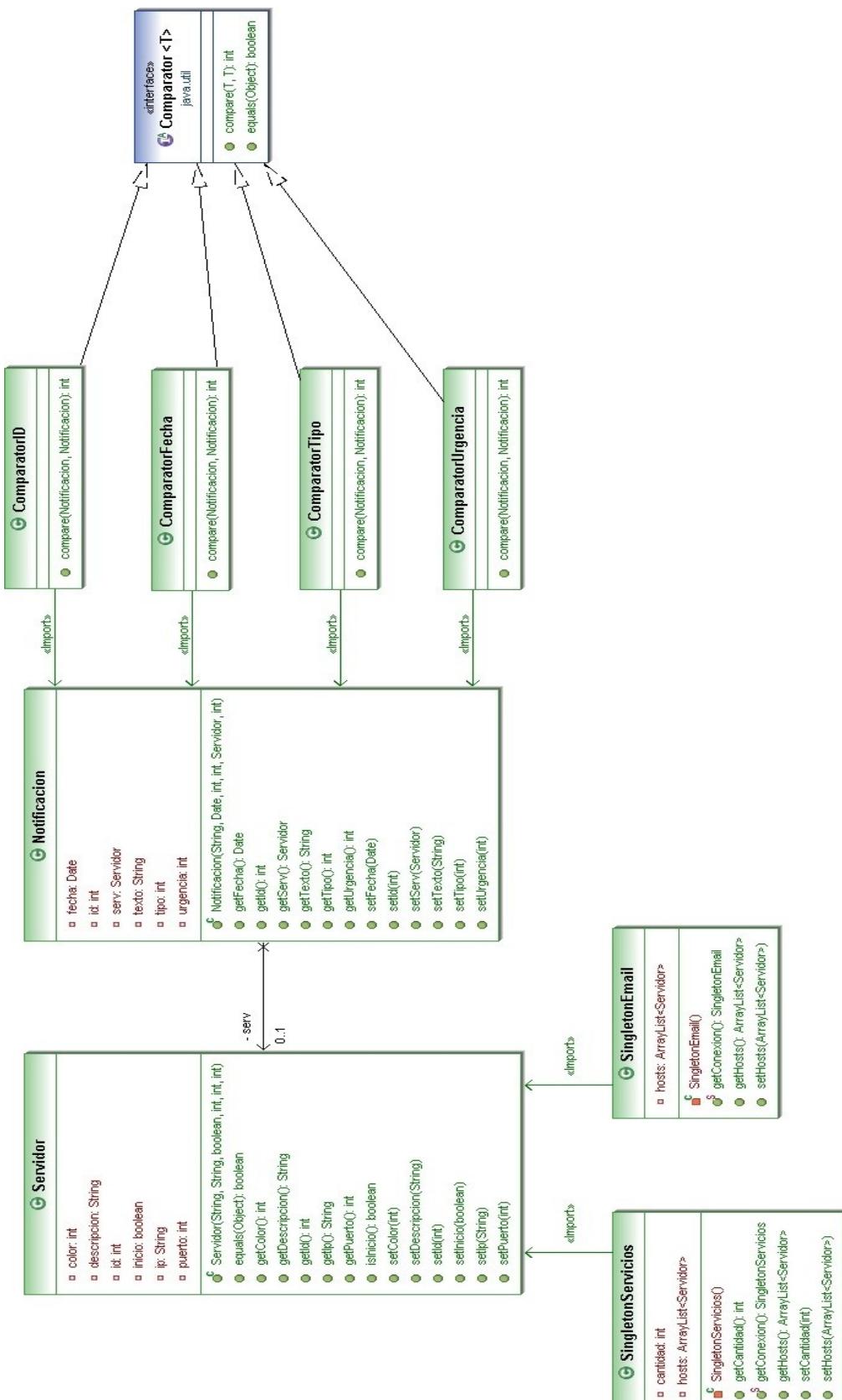
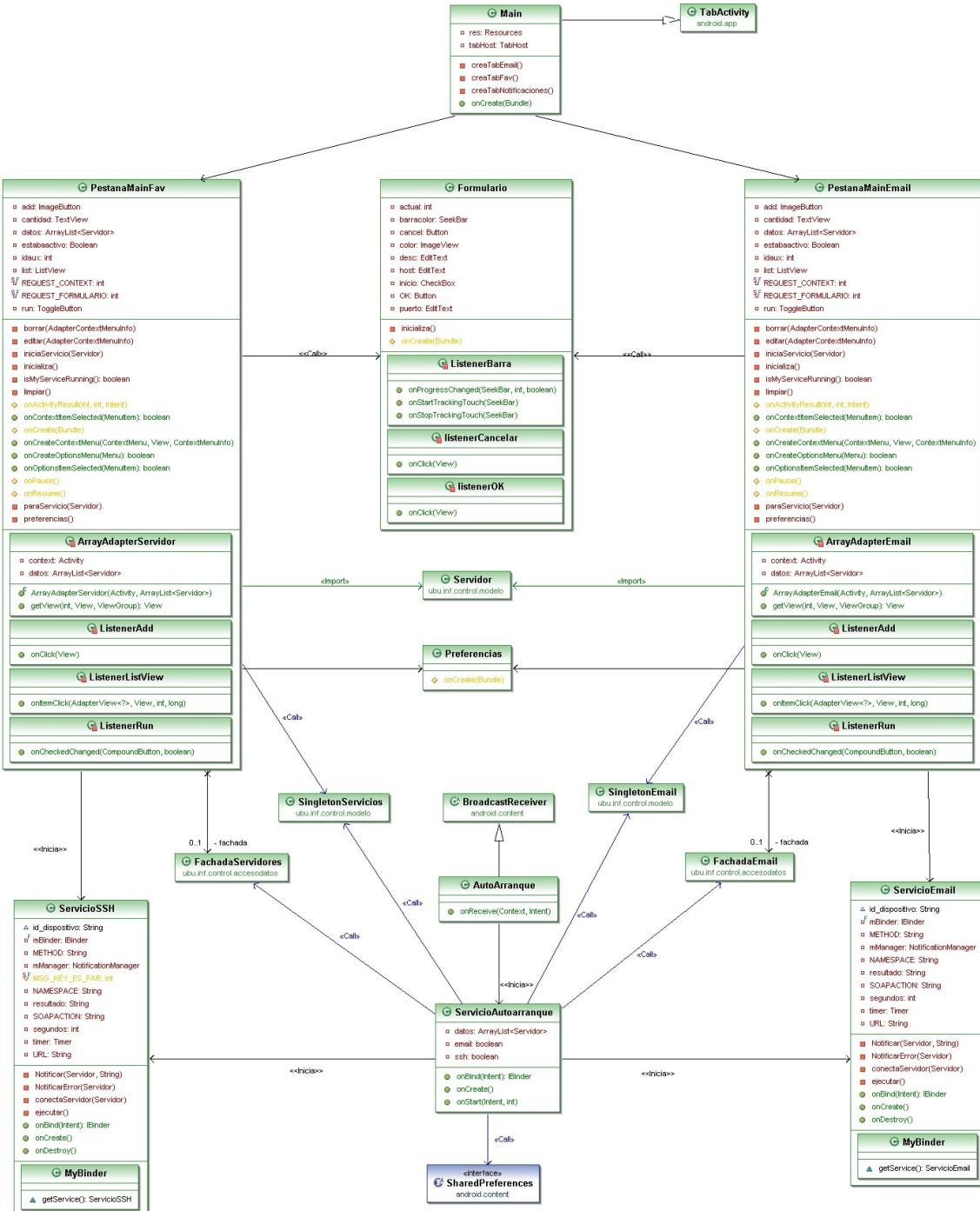


Ilustración 35: modulocontrol modelo

## Paquete accesodatos:

Como el diagrama de clases es muy grande, se ha dividido en dos:

## Parte 1:



*Ilustración 36: modulocontrol logical*

## Parte 2:

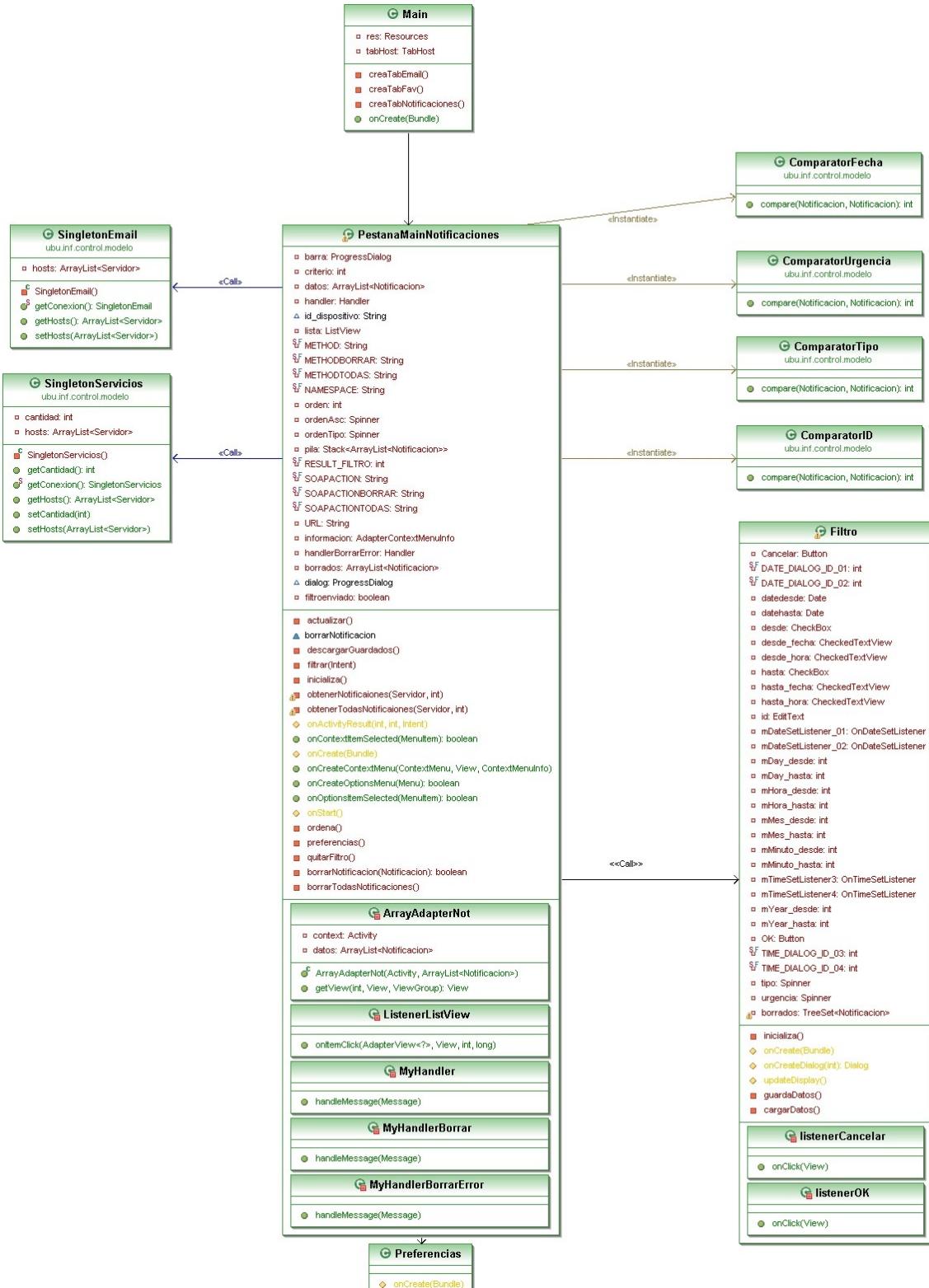


Ilustración 37: modulocontrol logica2

#### 4.5.4 Principal

Este módulo solo encarga de lanzar el resto de módulos y de controlar el acceso a los mismos.

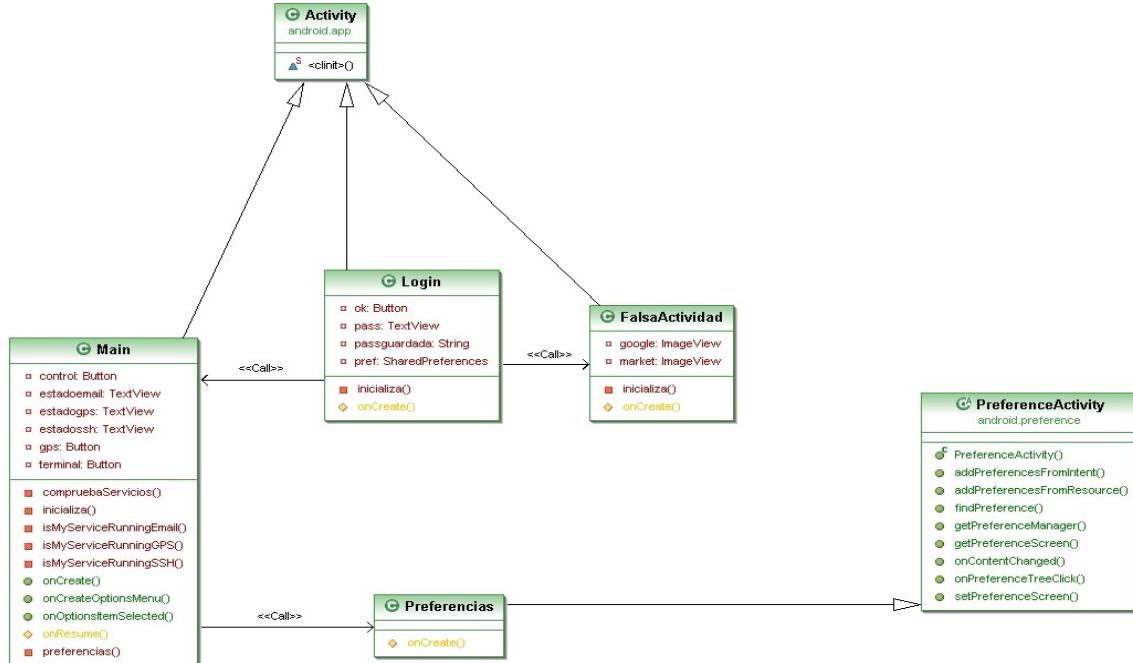


Ilustración 38: Principal

#### 4.5.5 Web Service

El Web Service se compone de dos partes como antes hemos dicho, el analizador de log y el Web Service.

Diagrama de clases del analizador de log:

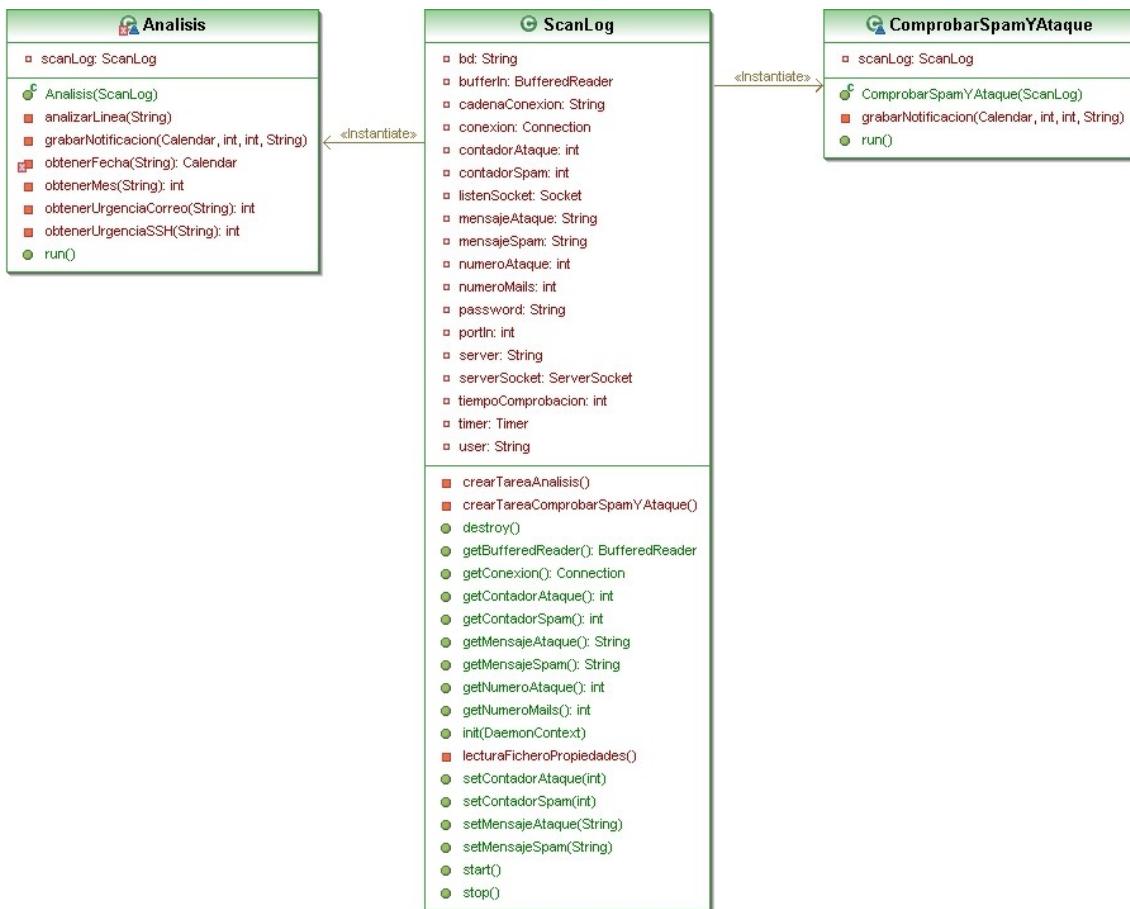


Ilustración 39: Web Service analizador

Diagrama de clases del Notificador (Web Service):



Ilustración 40: Web Service notificador

## 5 DISEÑO DE LA INTERFAZ

Como siempre, se diferenciará la parte del Web Service que se instala en el servidor y el cliente que se usará en Android.

En este caso el Web Service no dispone de interfaz gráfica, ya que toda la configuración se realiza mediante archivos de configuración y no necesita mostrar ningún dato.

Por su parte, en el cliente Android, uno de los mayores requisitos es el presentar los datos de forma ordenada, sencilla e intuitiva para el usuario, y es por eso que la interfaz gráfica cobra especial importancia.

Todas las aplicaciones Android que hemos desarrollado tendrán una pantalla común, la de ajustes, y que sigue el mismo formato que las pantallas de ajustes de Android.

A continuación representaremos de forma esquemática el diseño que debe tener la interfaz de los distintos módulos, es posible que se necesiten más de una interfaz para cada módulo, en cuyo caso se indicará.

### 5.1 Principal

Esta actividad actúa como launcher del resto, además de ejercer de control de acceso mediante el uso de una contraseña. Además cuenta con una pantalla falsa a la que se llegará cuando la contraseña sea incorrecta.

### 5.1.1 Pantalla de login

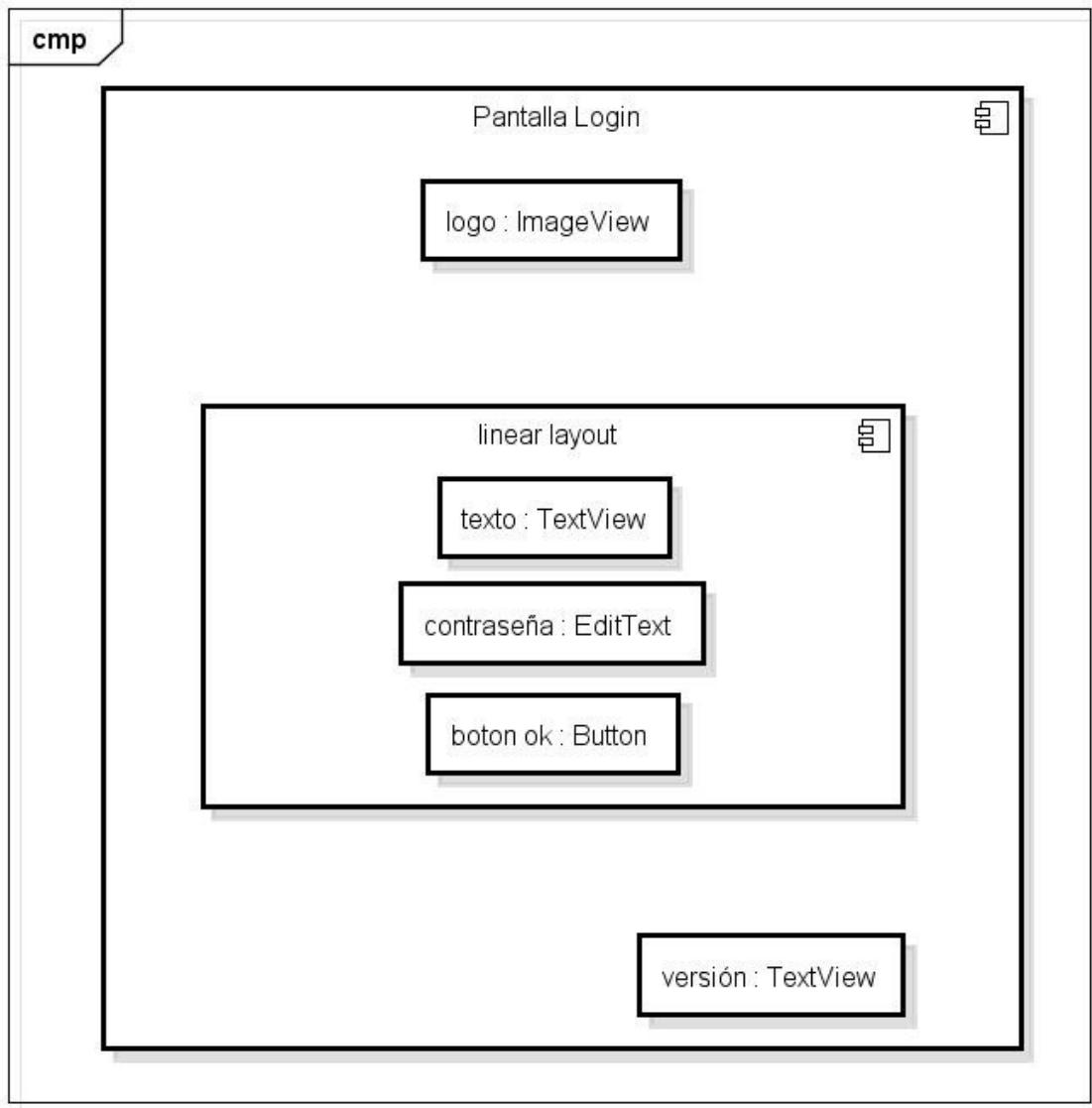


Ilustración 41: Principal login

### 5.1.2 Pantalla launcher

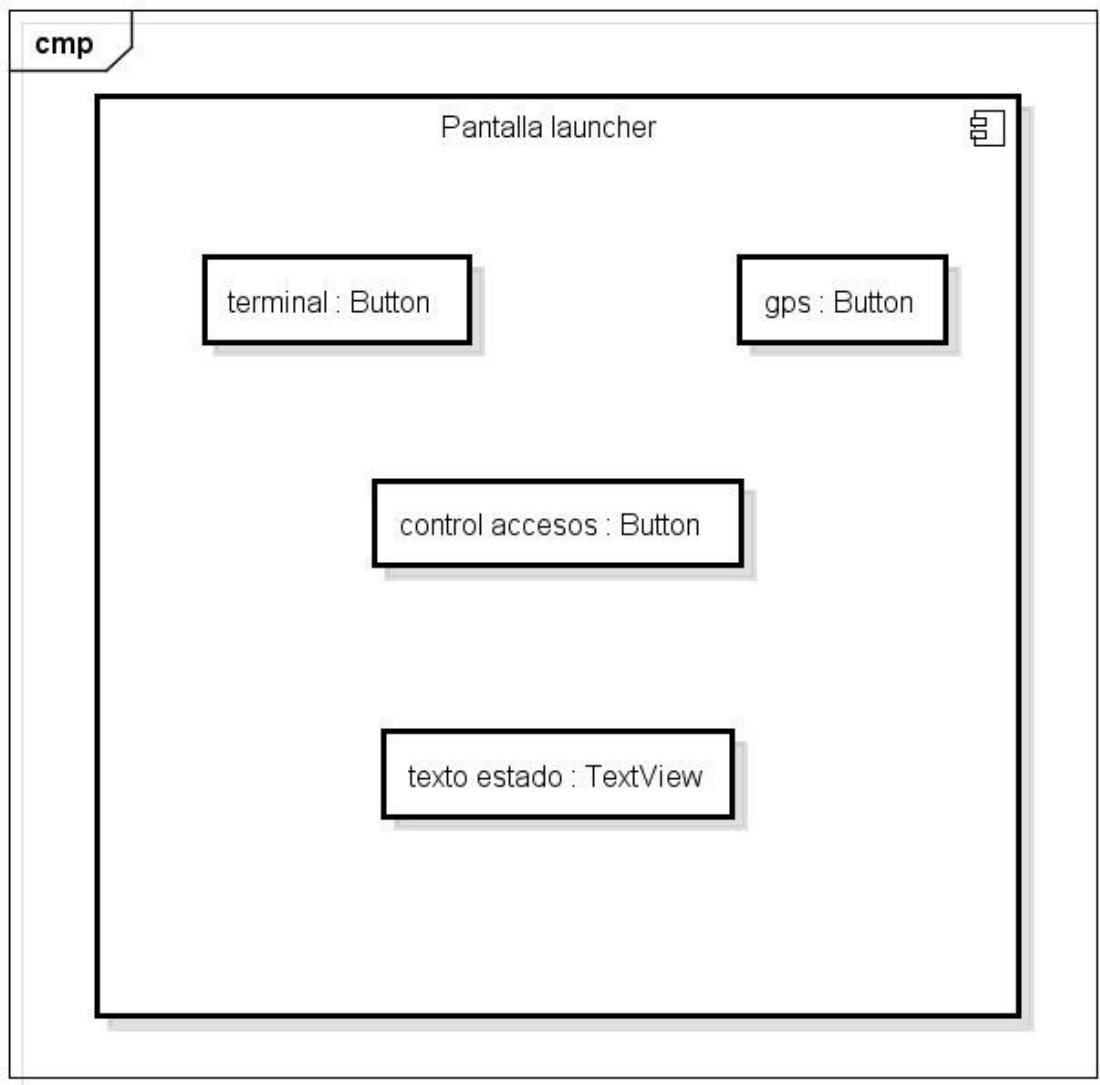


Ilustración 42: Principal launcher

### 5.1.3 Pantalla falsa

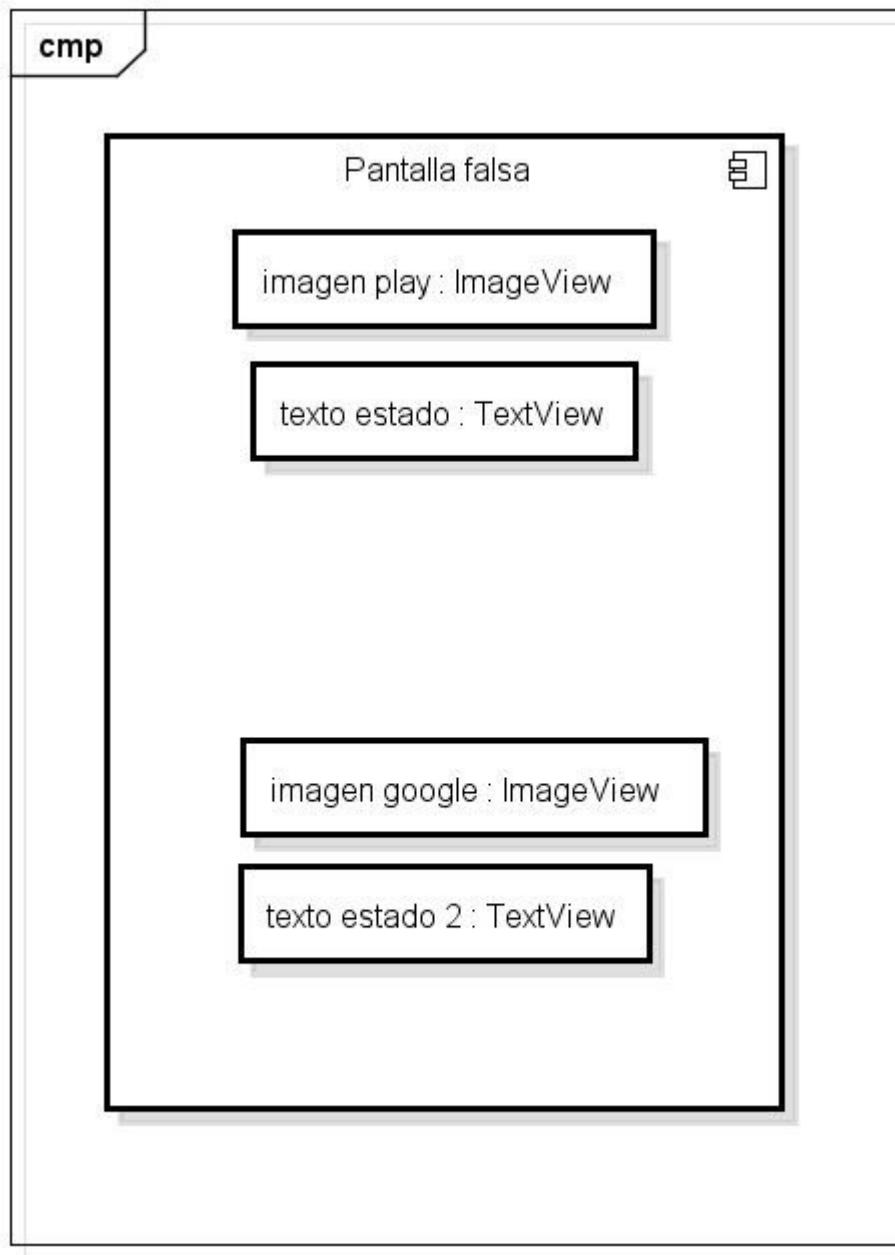


Ilustración 43: Principal falsa

## 5.2 Módulo Consola

Esta aplicación tiene varias pantallas, siendo las más importantes la pantalla principal desde la que se elige a qué servidor conectarse y la pantalla que emula a una terminal ssh una vez establecida la conexión.

En los diagramas se muestra la composición básica de la interfaz, en el caso de que

no entran todos los componentes gráficos se redimensionan, o se incluyen dentro de un **ScrollView** el cual permite hacer scroll por la pantalla, y que la interfaz contenga más componentes de los que caben en la pantalla.

### 5.2.1 Pantalla principal

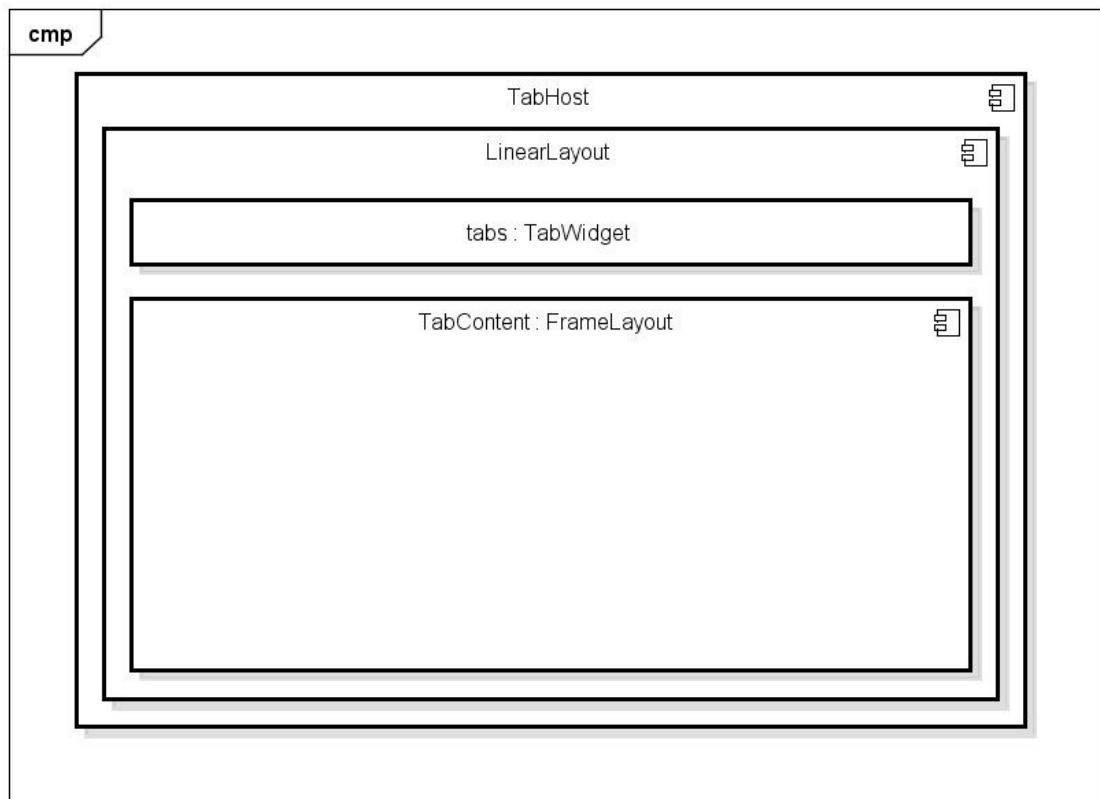


Ilustración 44: ssh principal

### 5.2.2 Pestaña manual

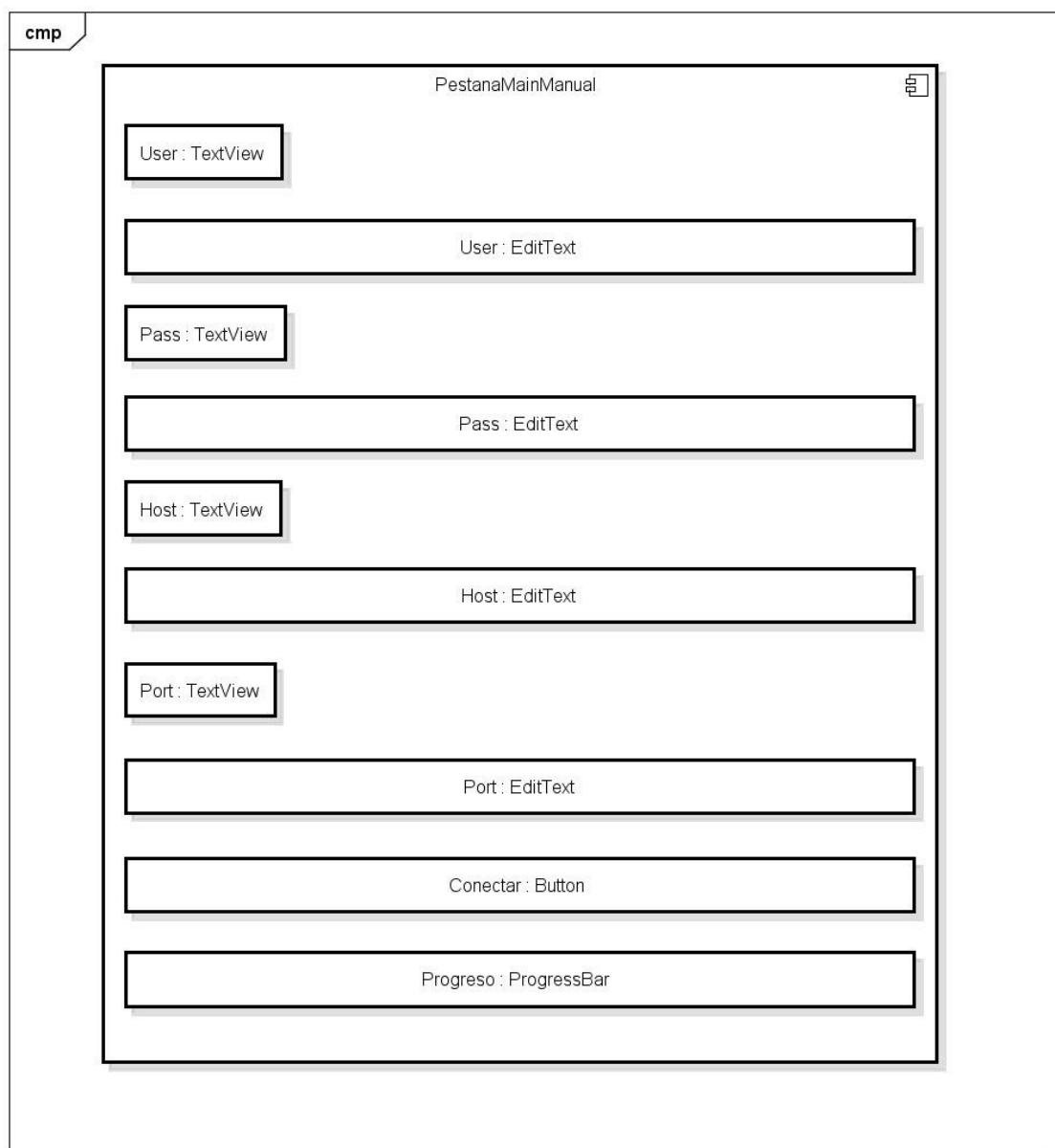


Ilustración 45: ssh pestaña manual

### 5.2.3 Pestaña favoritos

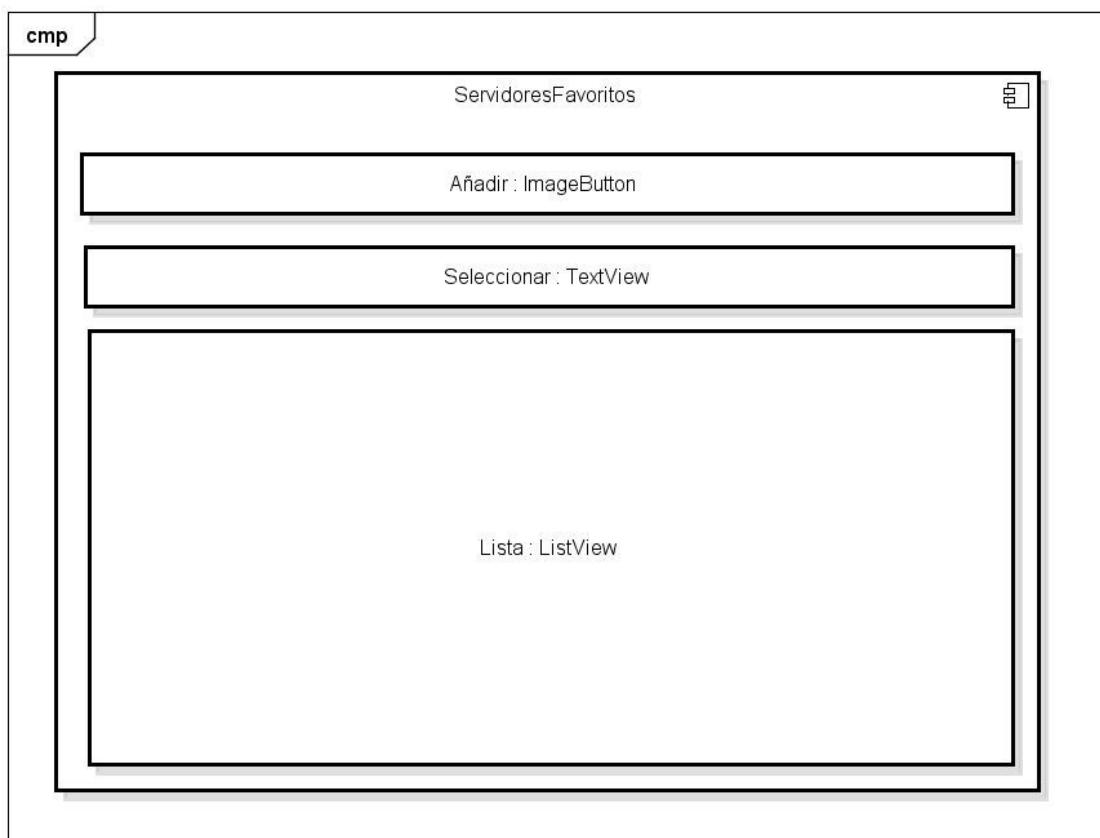


Ilustración 46: ssh servidores favoritos

### 5.2.4 Formulario

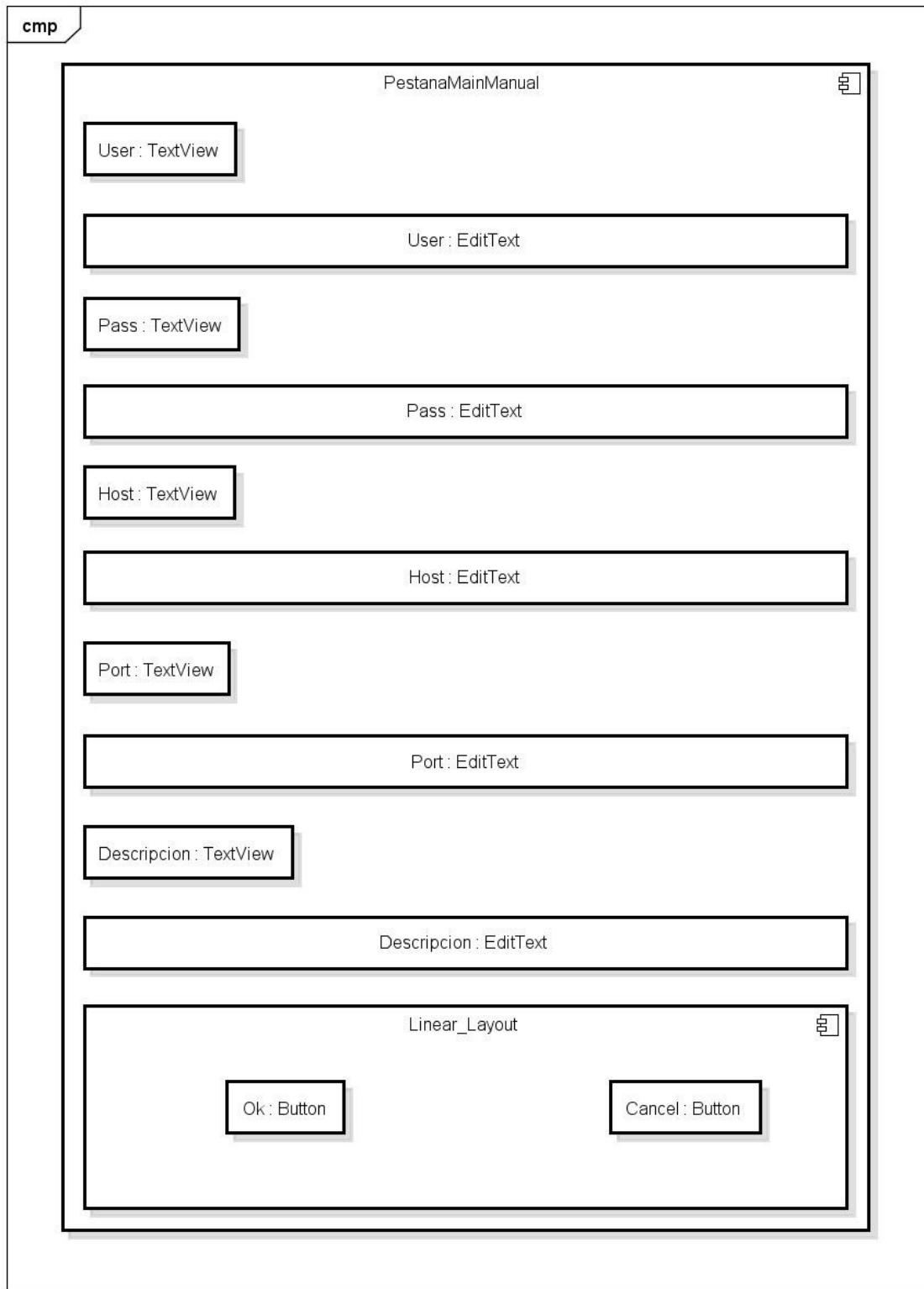


Ilustración 47: ssh formulario

### 5.2.5 Terminal SSH



Ilustración 48: ssh consola

### 5.2.6 Pantalla comandos ejecutados

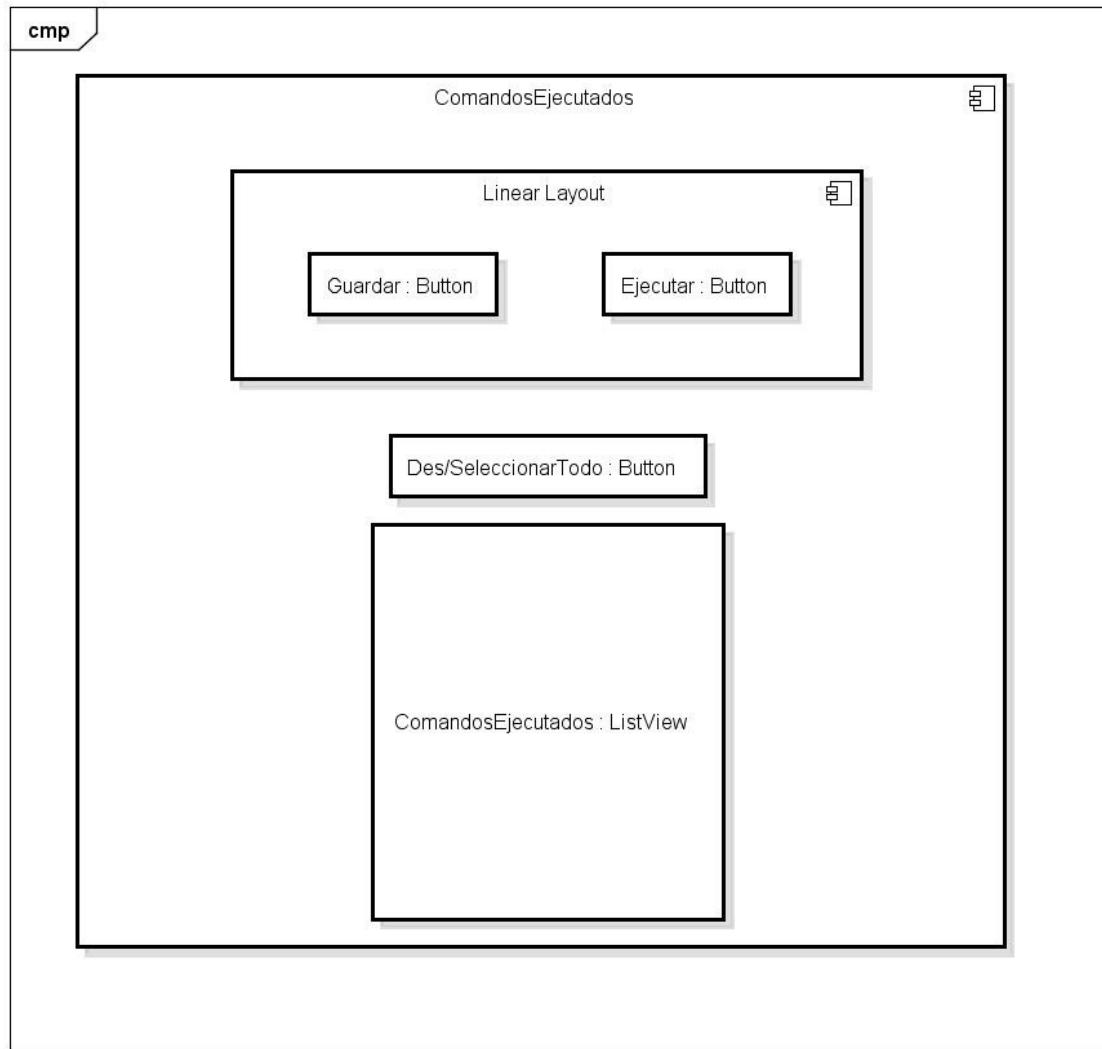


Ilustración 49: ssh comandos ejecutados

### 5.2.7 Pantalla comandos favoritos

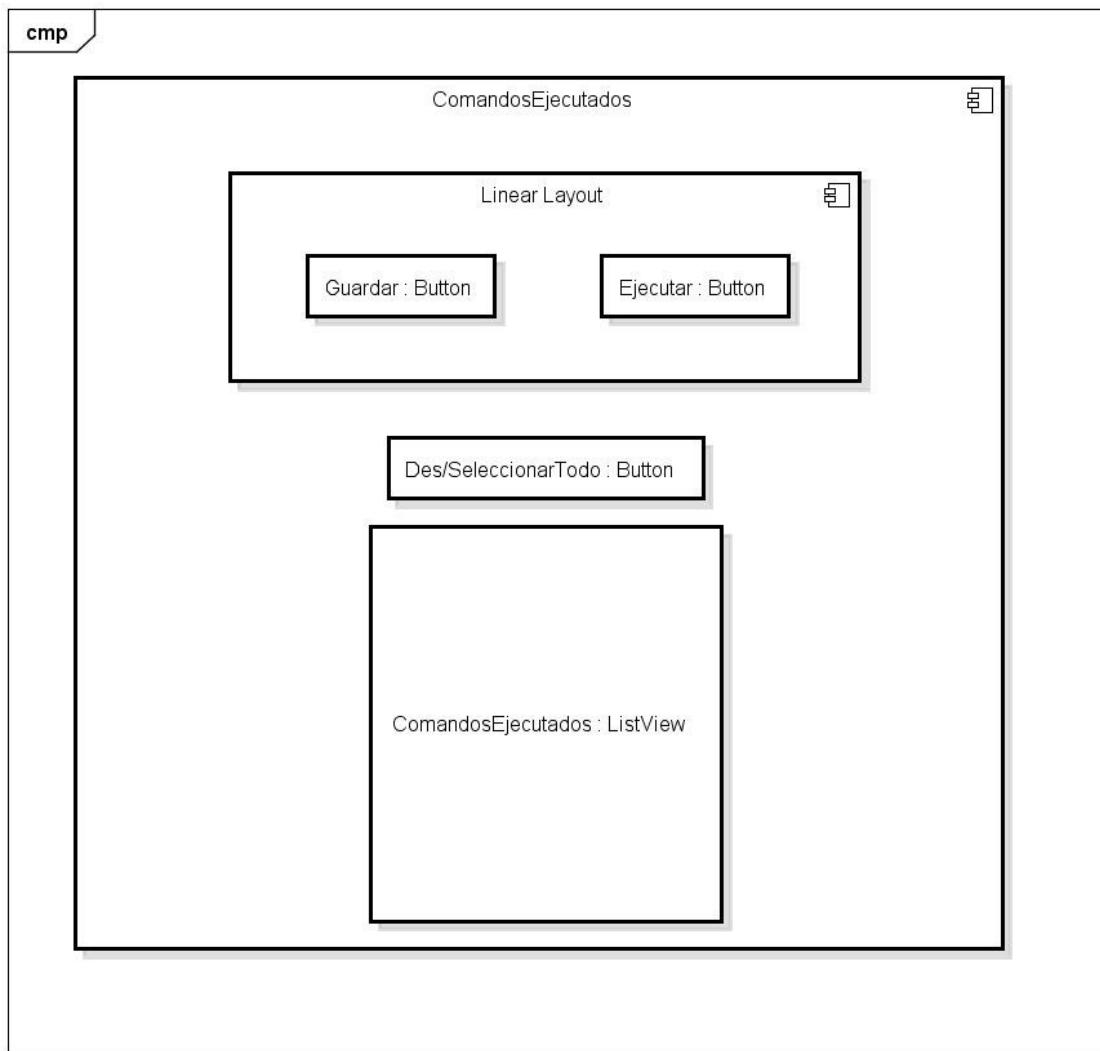


Ilustración 50: ssh comandos favoritos

## 5.3 Módulo GPS

### 5.3.1 Pantalla principal

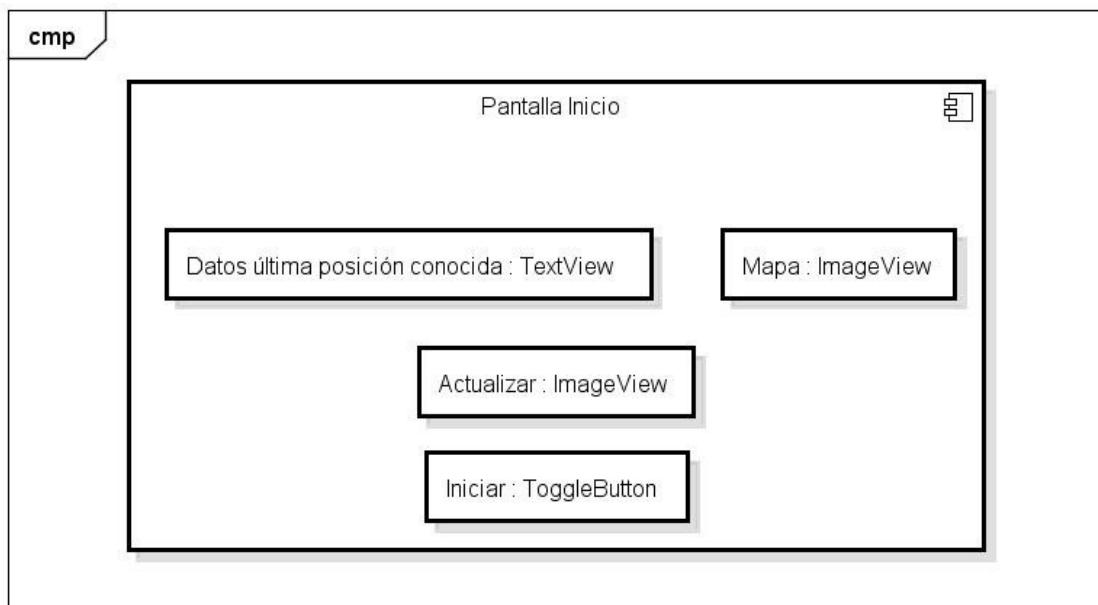


Ilustración 51: gps pantalla principal

## 5.4 Módulo Control de Accesos

### 5.4.1 Filtro

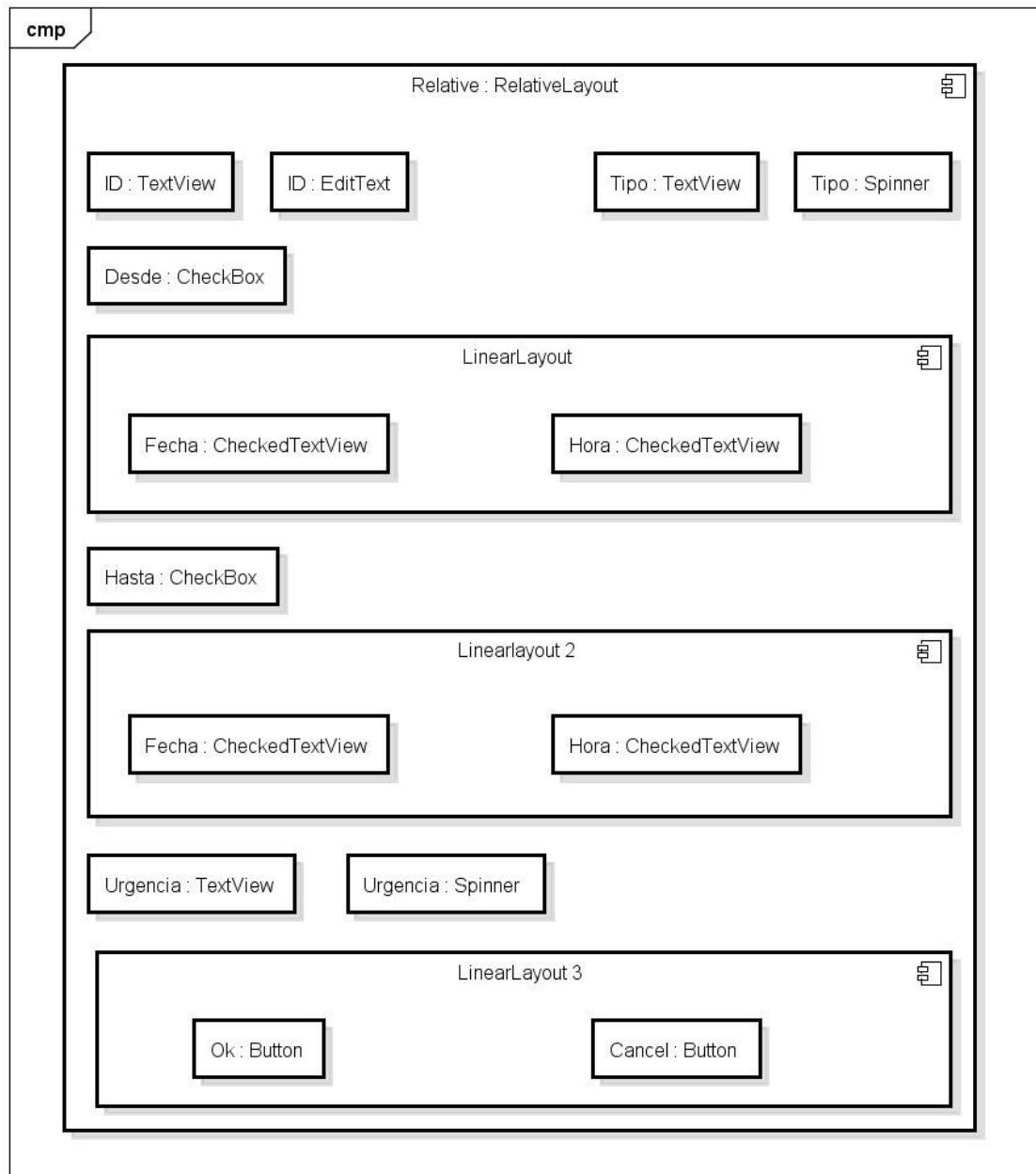


Ilustración 52: control filtro

### 5.4.2 Formulario

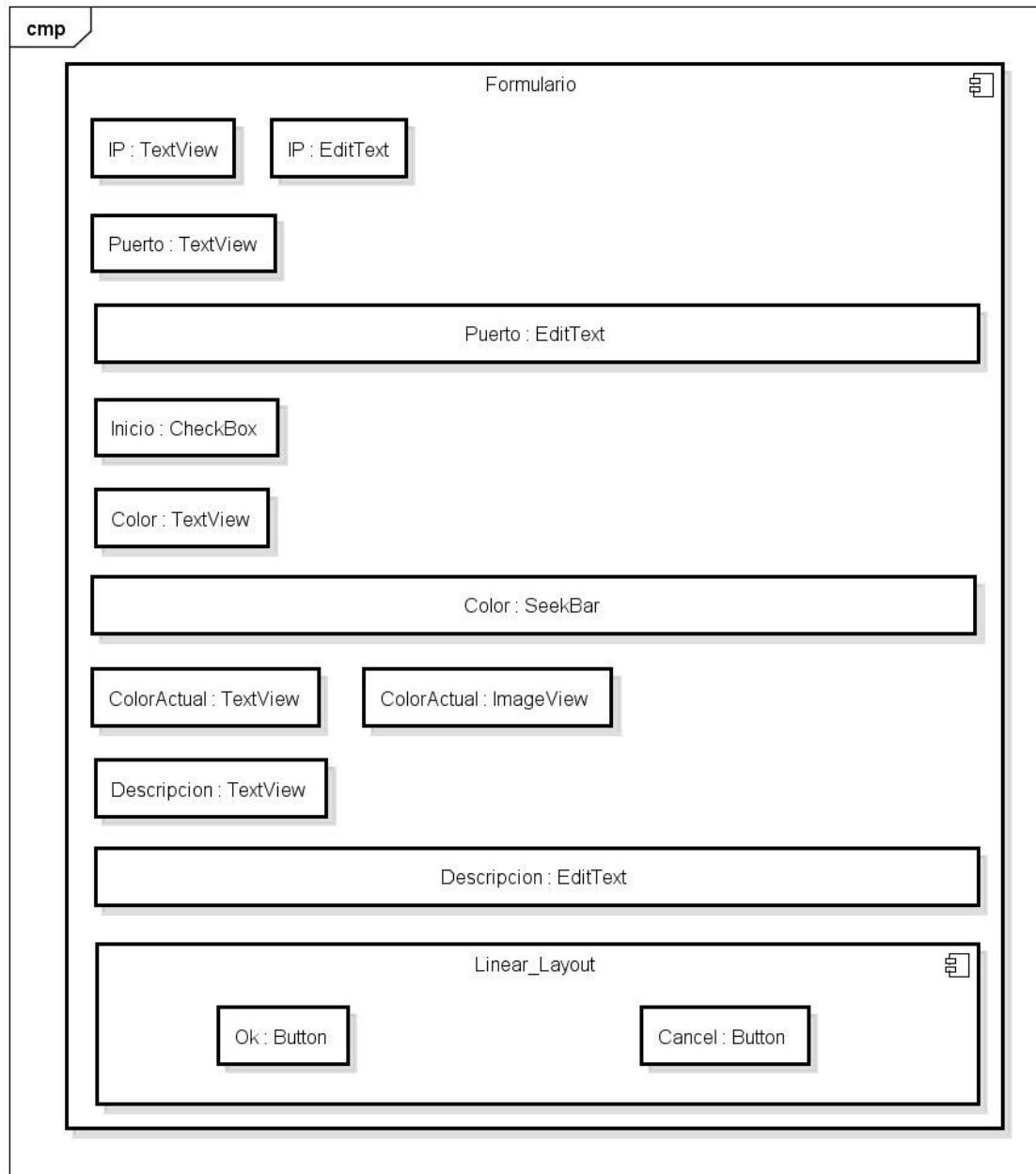


Ilustración 53: control formulario

### 5.4.3 Main

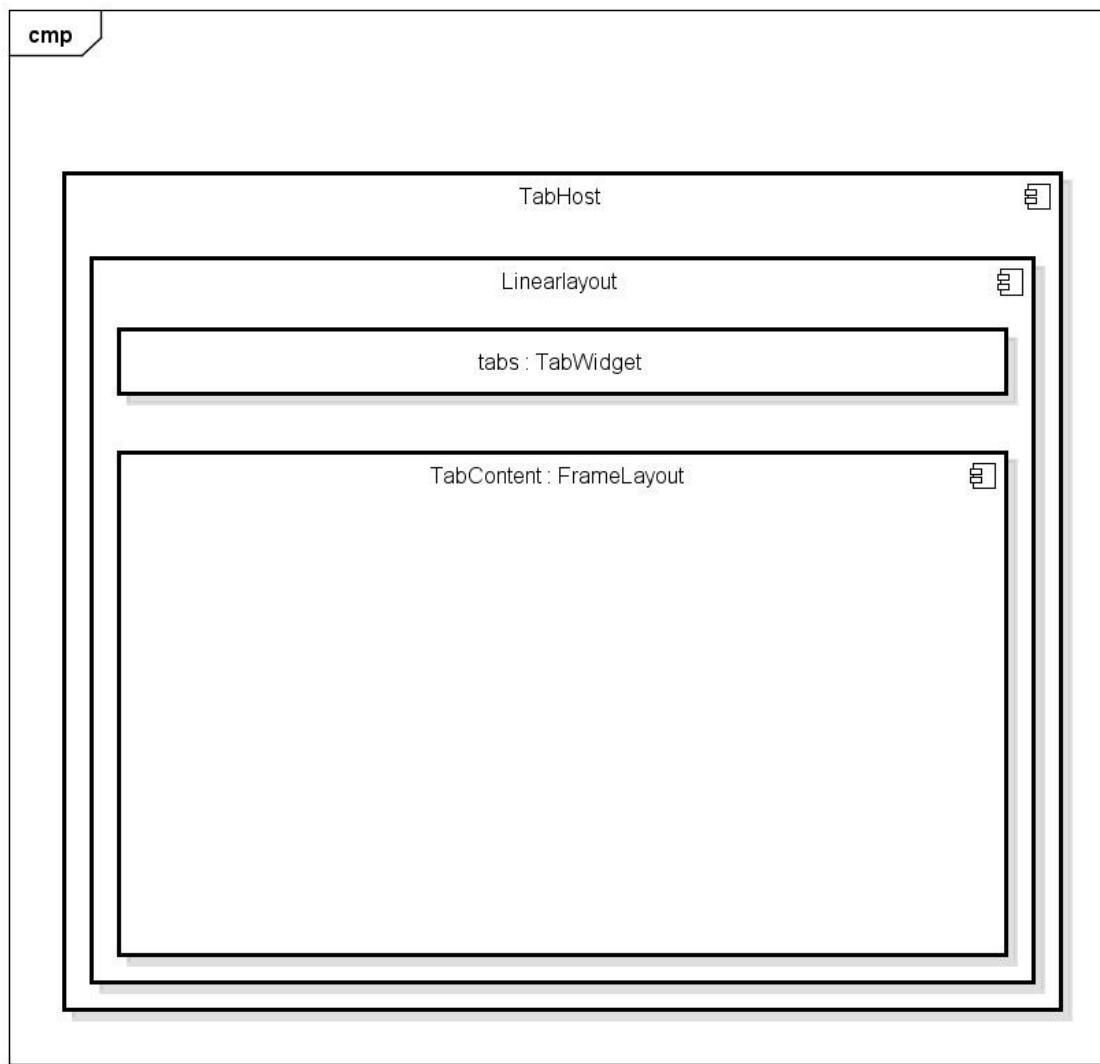


Ilustración 54: control main

#### 5.4.4 Pestaña favoritos

La pestaña con los servidores favoritos ssh y email son exactamente iguales.

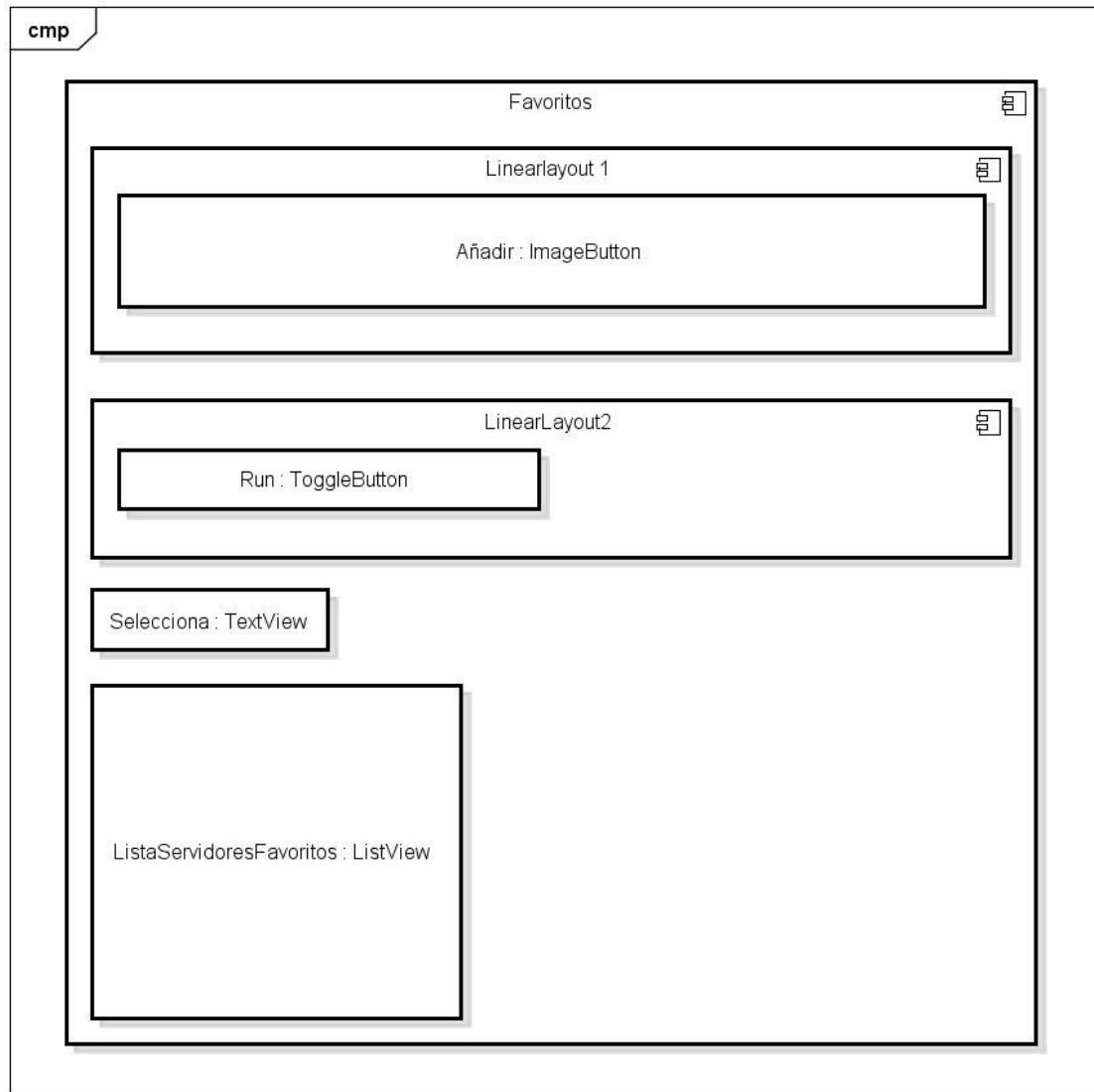


Ilustración 55: control pestaña

#### 5.4.4.1 Lista servidores

La lista que muestra los servidores tienen una interfaz propia, como todas las que se han usado en el proyecto, pero se ha decidido incorporar el diagrama de interaz de esta debido a su complejidad.

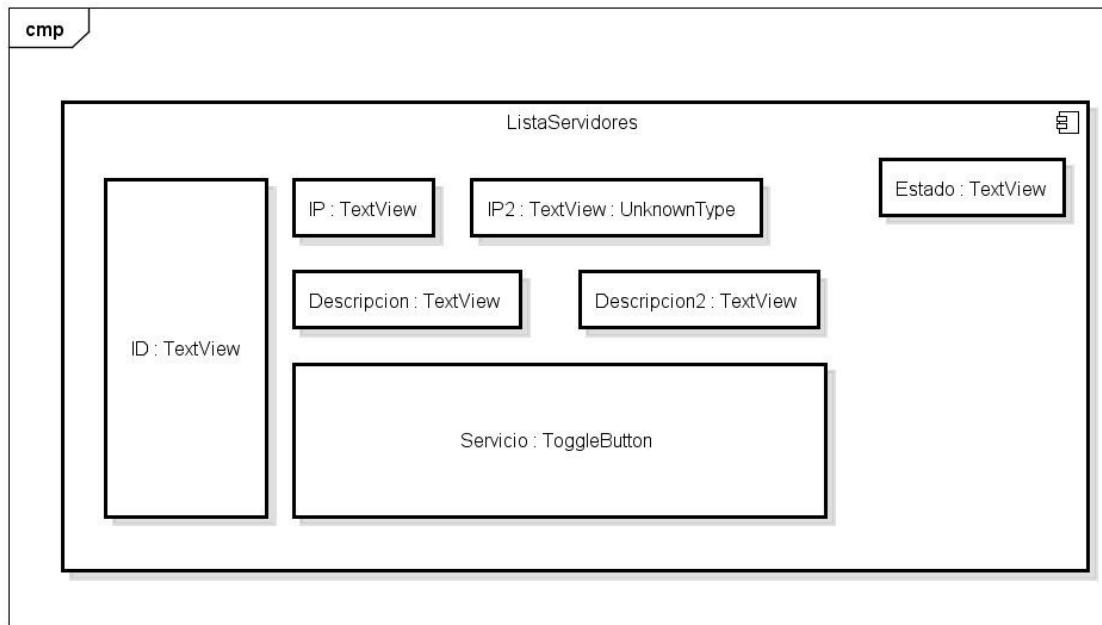


Ilustración 56: control lista servidores

### 5.4.5 Pestaña notificaciones

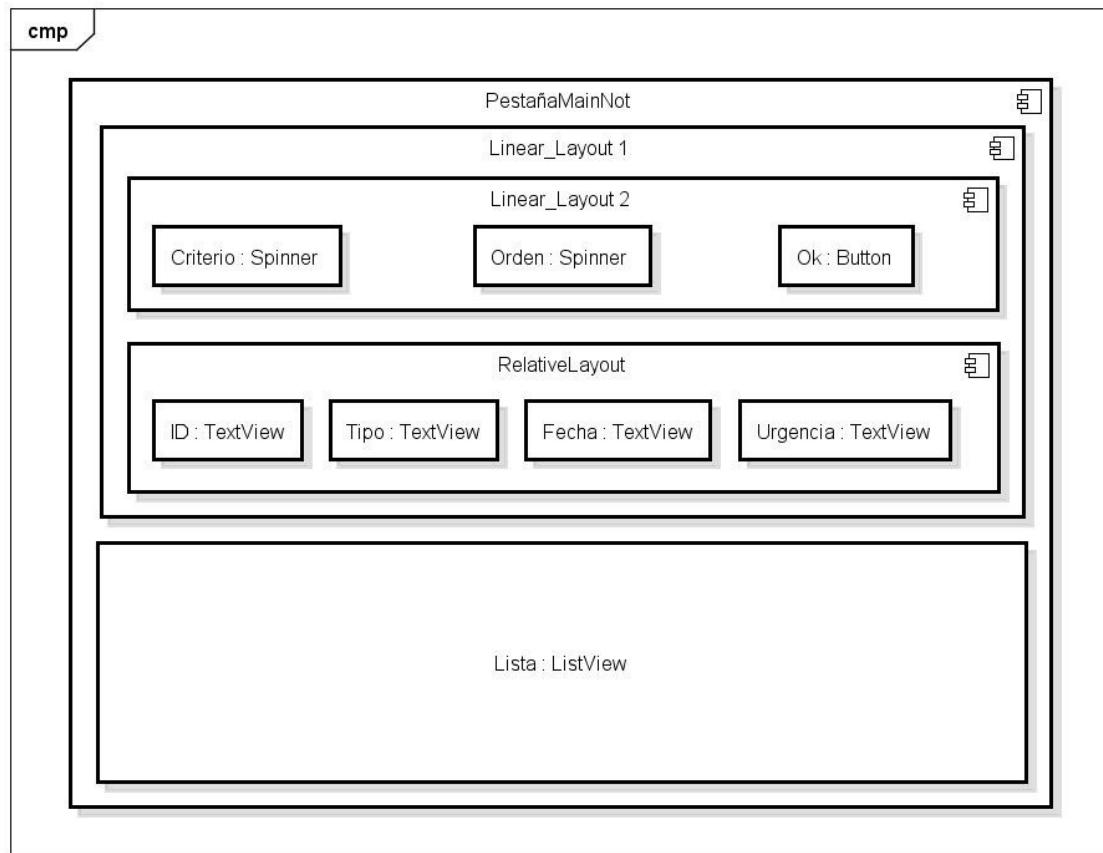


Ilustración 57: control pestaña notificaciones

## 6 DISEÑO PROCEDIMENTAL

A continuación se detallan los diagramas de secuencias de las operaciones más importantes o aquellas que pueden dar lugar a confusión.

Se ha intentado que los diagramas sean lo más fieles posibles a la funcionalidad de la aplicación, aunque en algunos casos se ha decidido simplificar los diagramas para una mayor claridad.

Como siempre se dividirá en apartados, uno por cada módulo de la aplicación.

## 6.1 Principal

### 6.1.1 Login

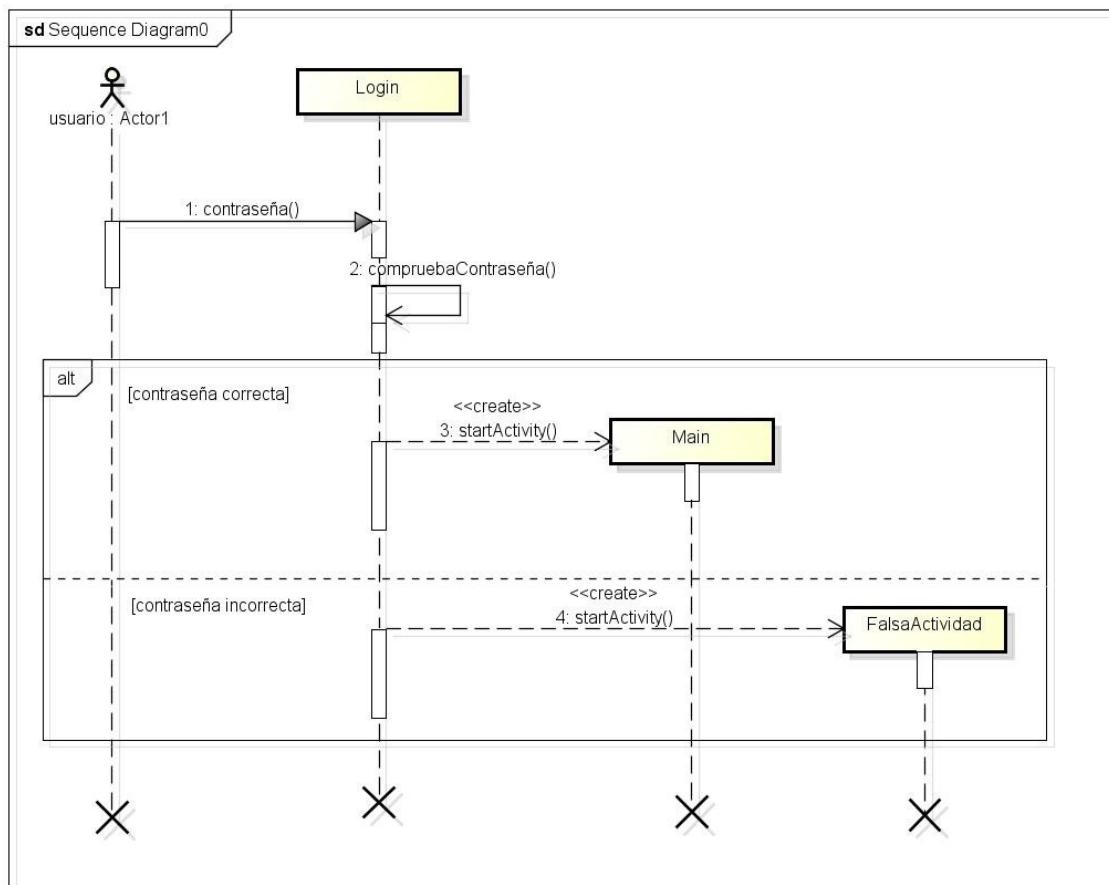


Ilustración 58: Secuencia principal login

### 6.1.2 Main

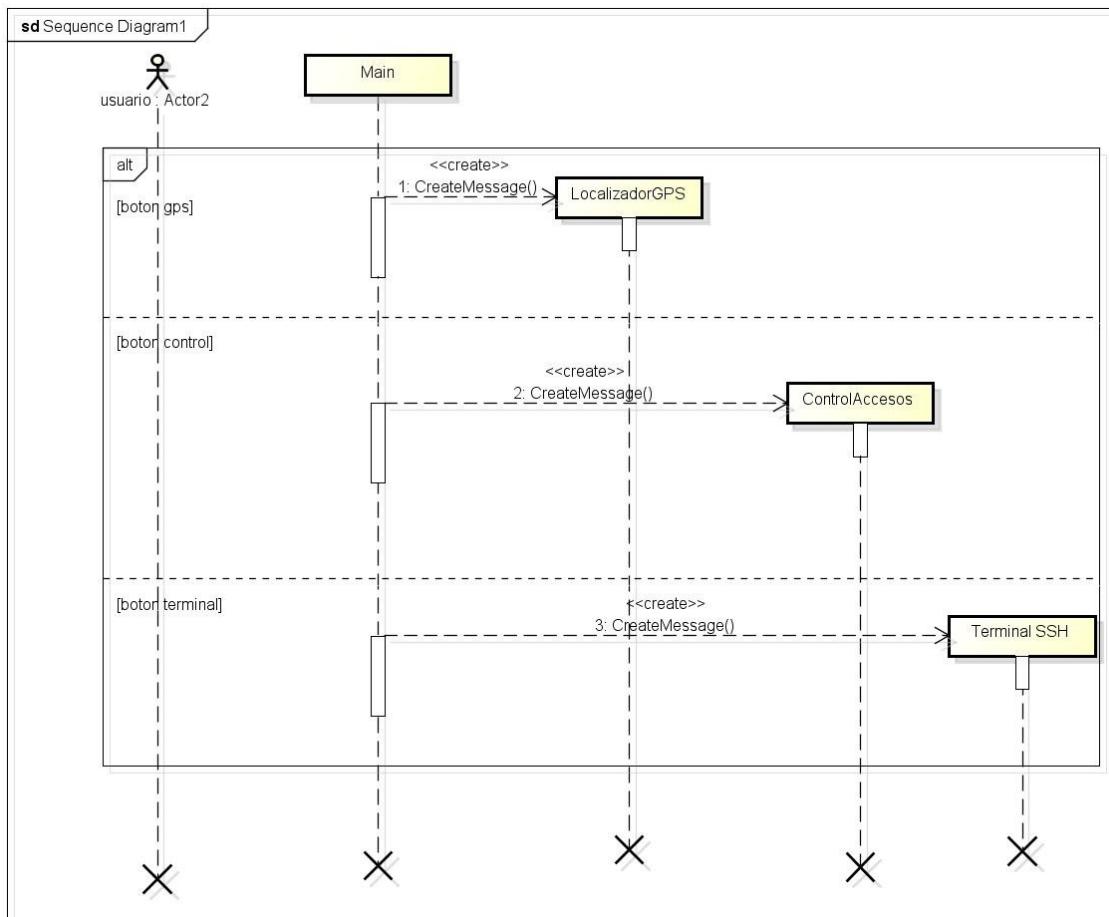


Ilustración 59: principal secuencia main

## 6.2 Módulo SSH

### 6.2.1 Guardar nuevo servidor

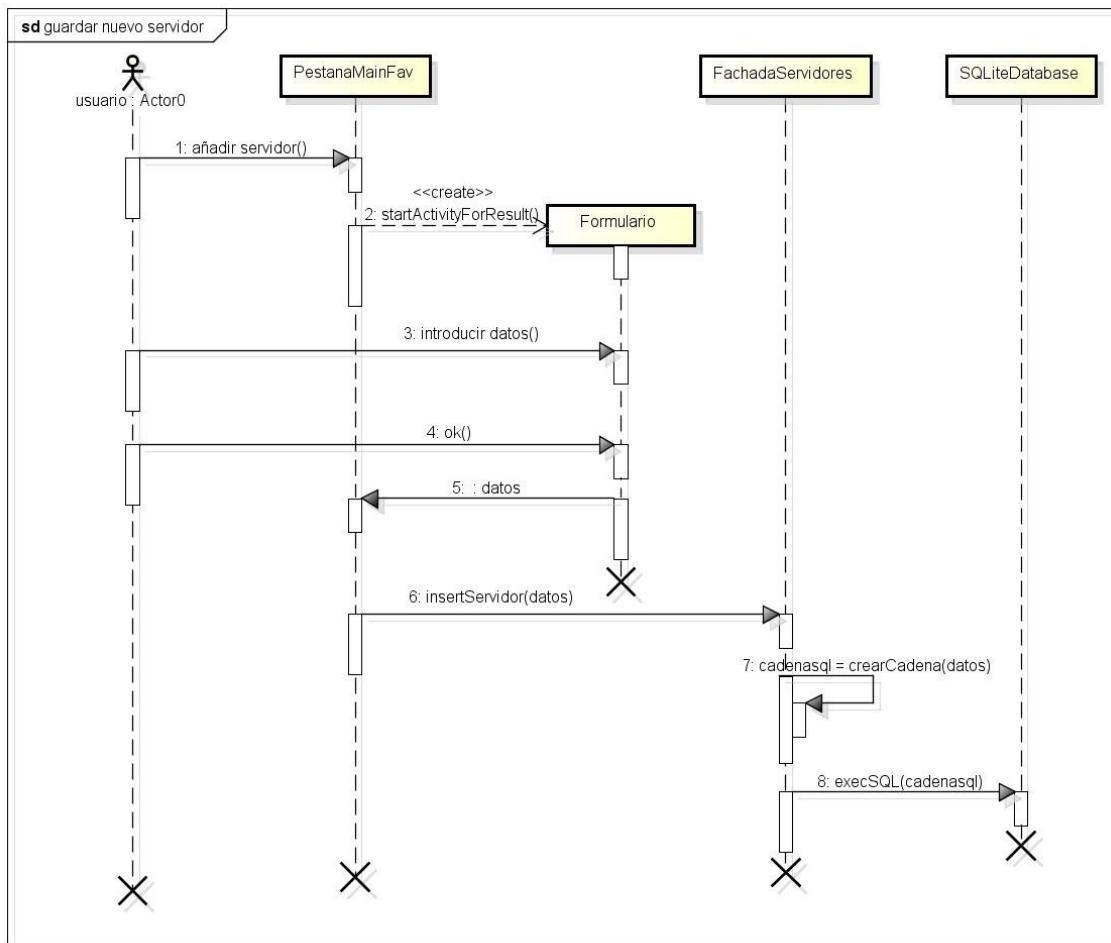


Ilustración 60: ssh nuevo servidor

### 6.2.2 Conectar a servidor

Al conectarse a un servidor, se puede hacer de forma manual o conectarse a un servidor guardado previamente, el modo de operar es el mismo en ambos casos, así que mostraremos el caso de que se conecta a un servidor ya guardado.

Cuando nos conectamos a un servidor, inmediatamente obtenemos los Streams de lectura y escritura para poder comunicarnos con él, además creamos un hilo asíncrono para poder leer del Stream sin que se bloquee la aplicación.

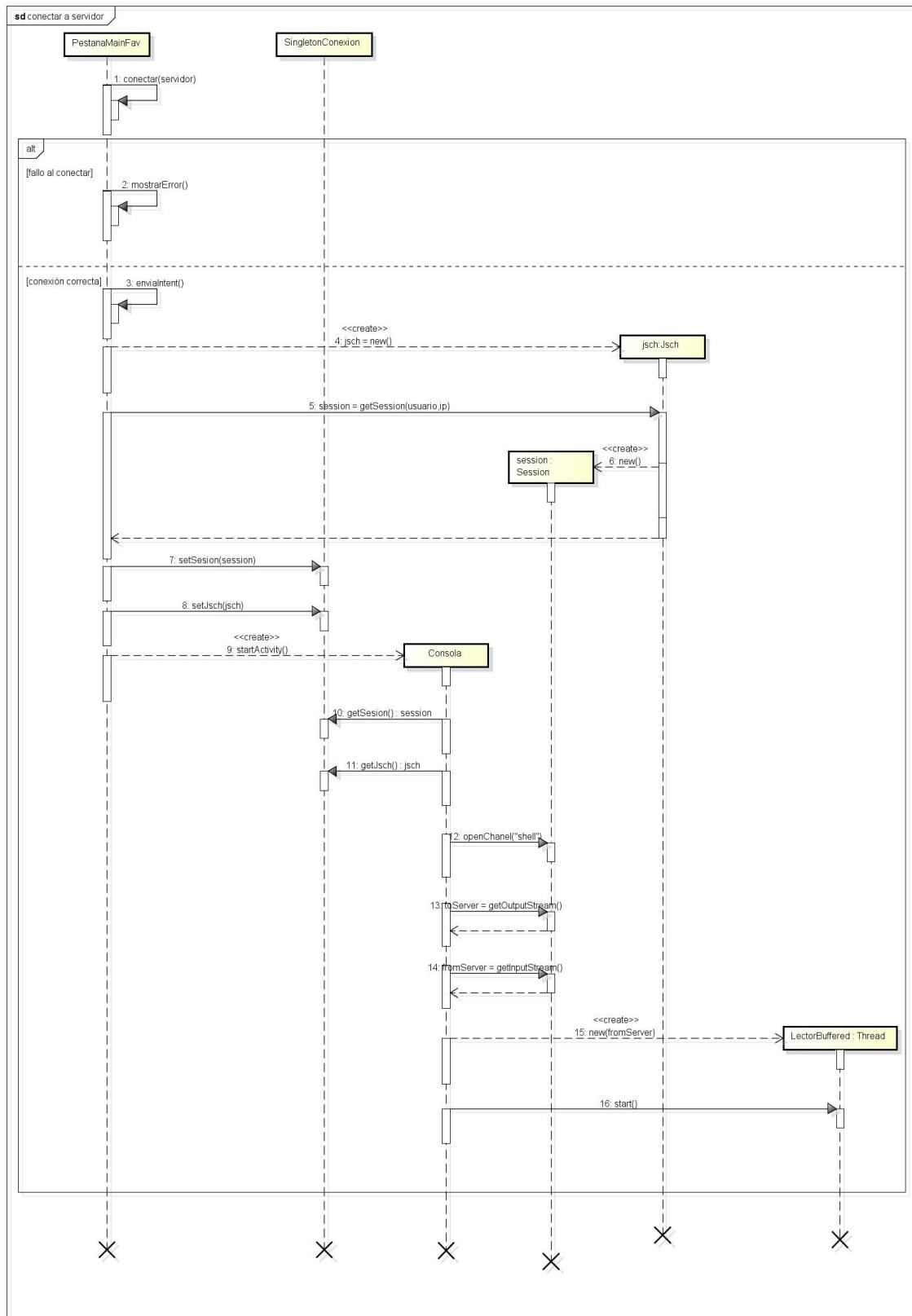


Ilustración 61: ssh conectar

### 6.2.3 Enviar comando

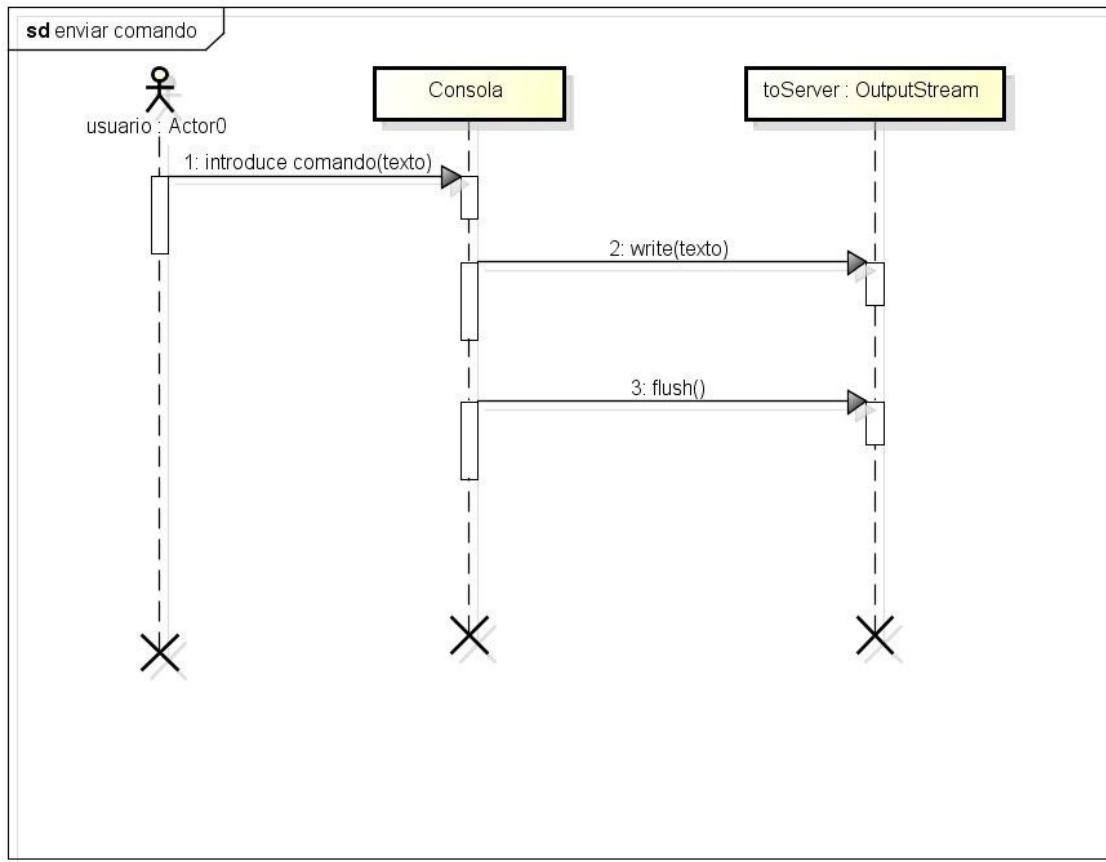


Ilustración 62: ssh enviar comando

### 6.2.4 Guardar script

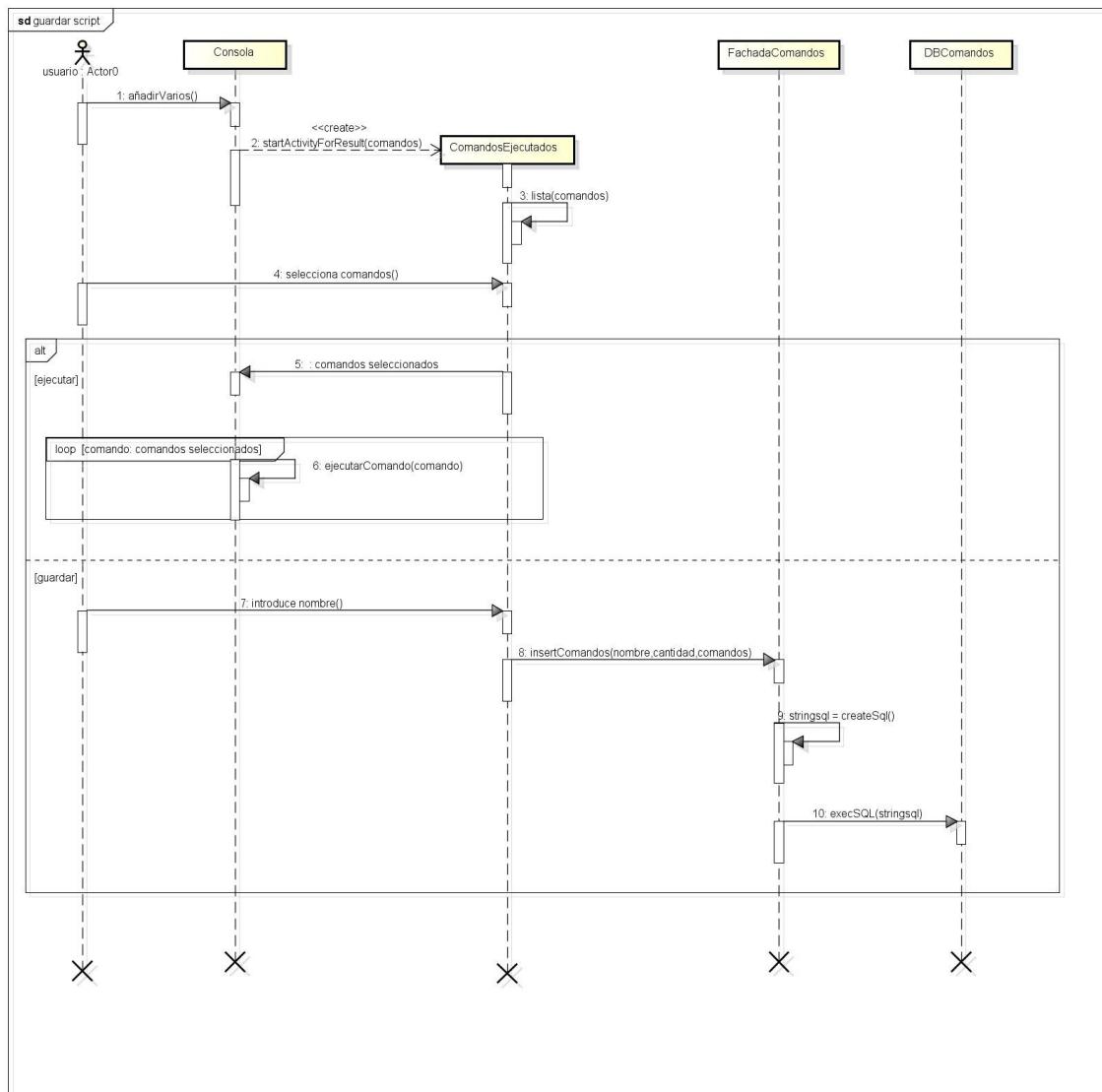


Ilustración 63: ssh guardar script

### 6.2.5 Ejecutar script guardado

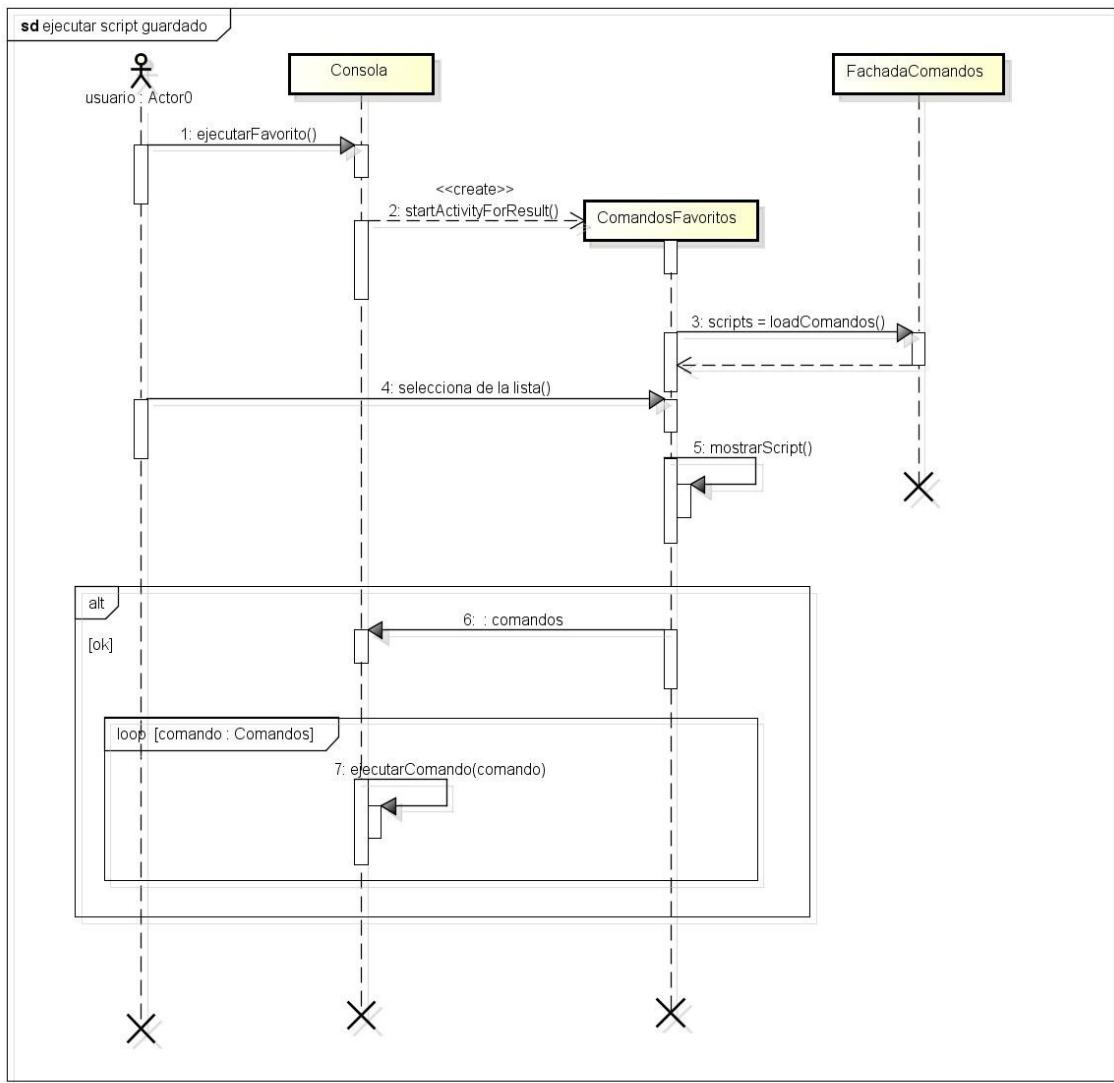


Ilustración 64: ssh ejecutar guardado

## 6.3 Módulo GPS

### 6.3.1 Iniciar desde sms

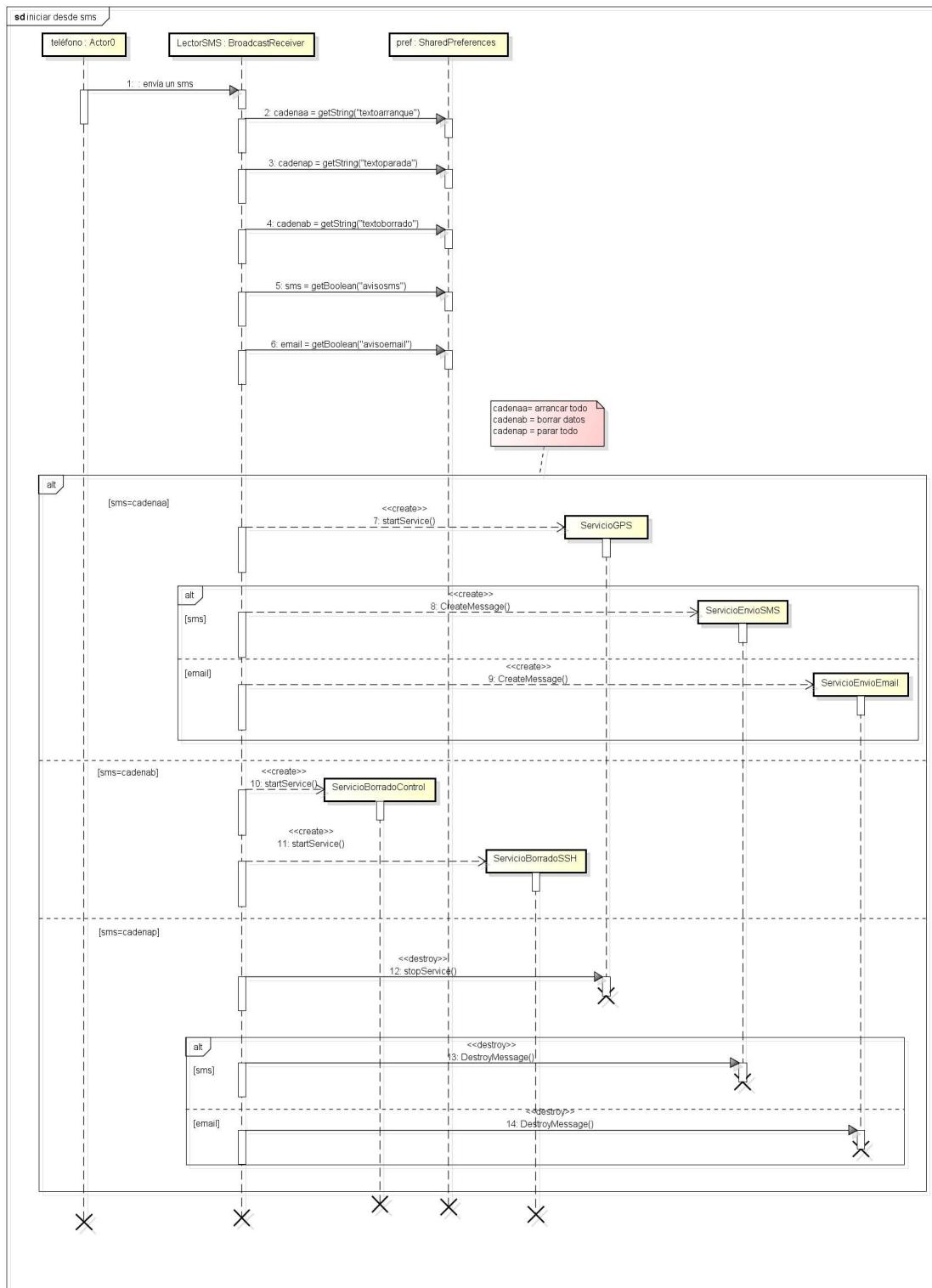


Ilustración 65: gps inicia desde sms

### 6.3.2 Guardar localizaciones

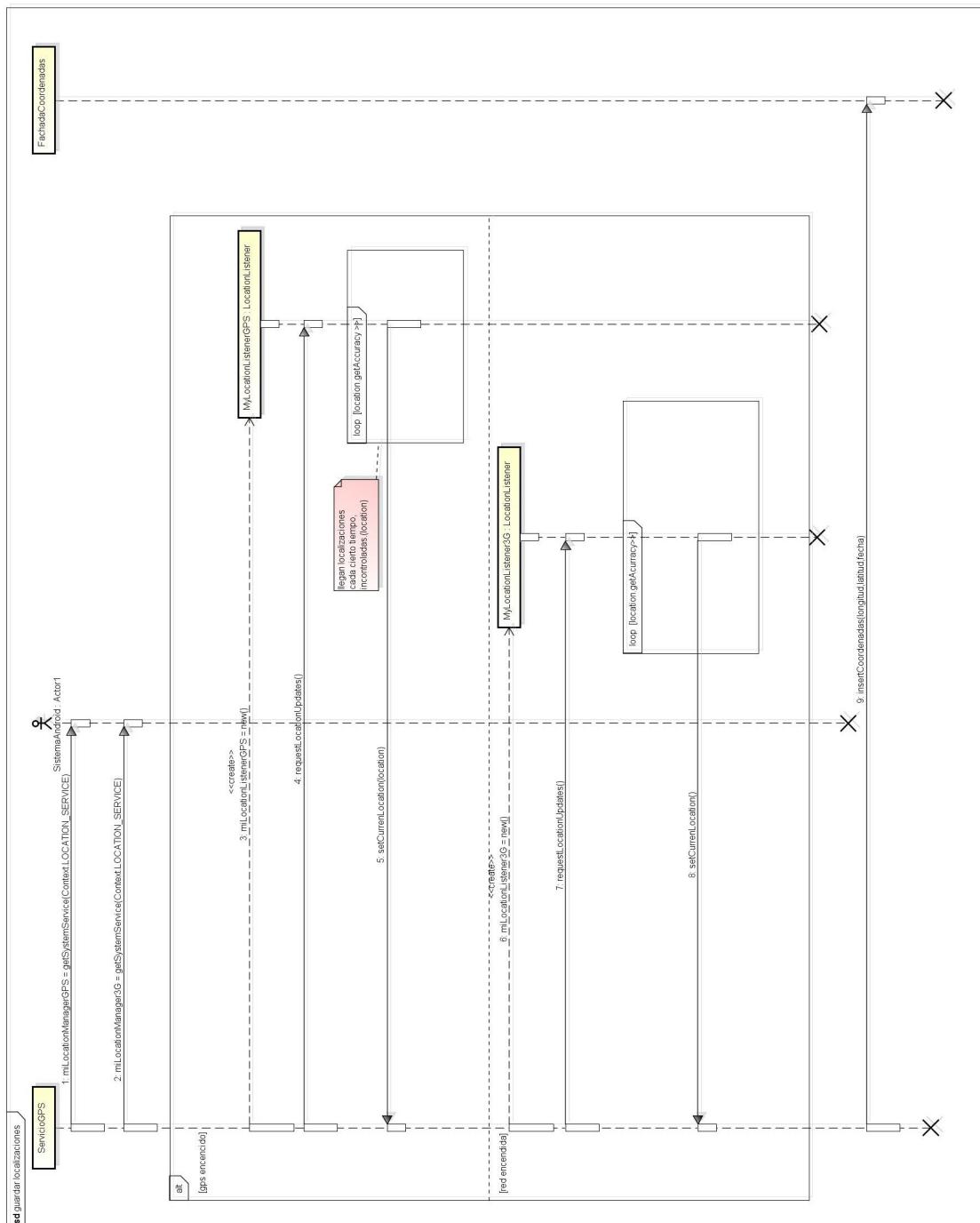


Ilustración 66: gps guardar localización

### 6.3.3 Enviar correo electrónico

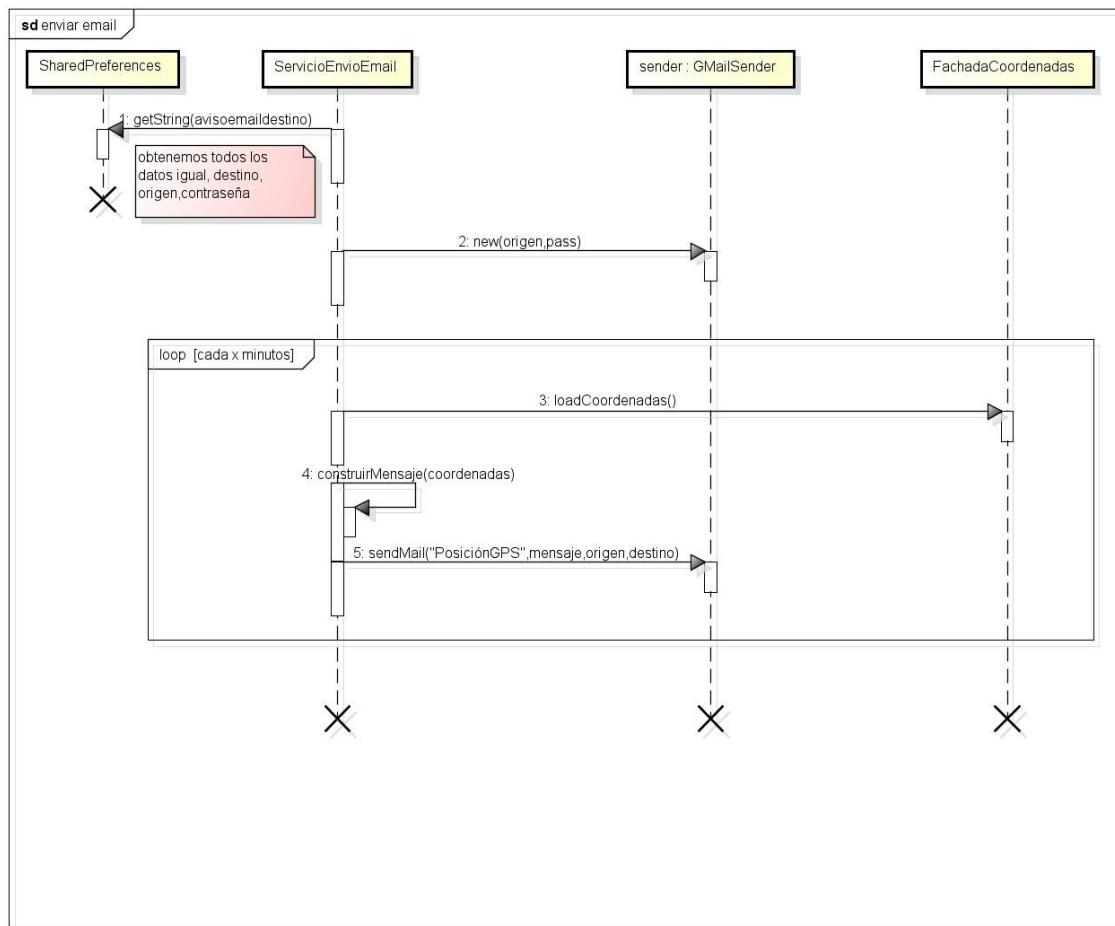


Ilustración 67: gps enviar correo electrónico

### 6.3.4 Enviar sms

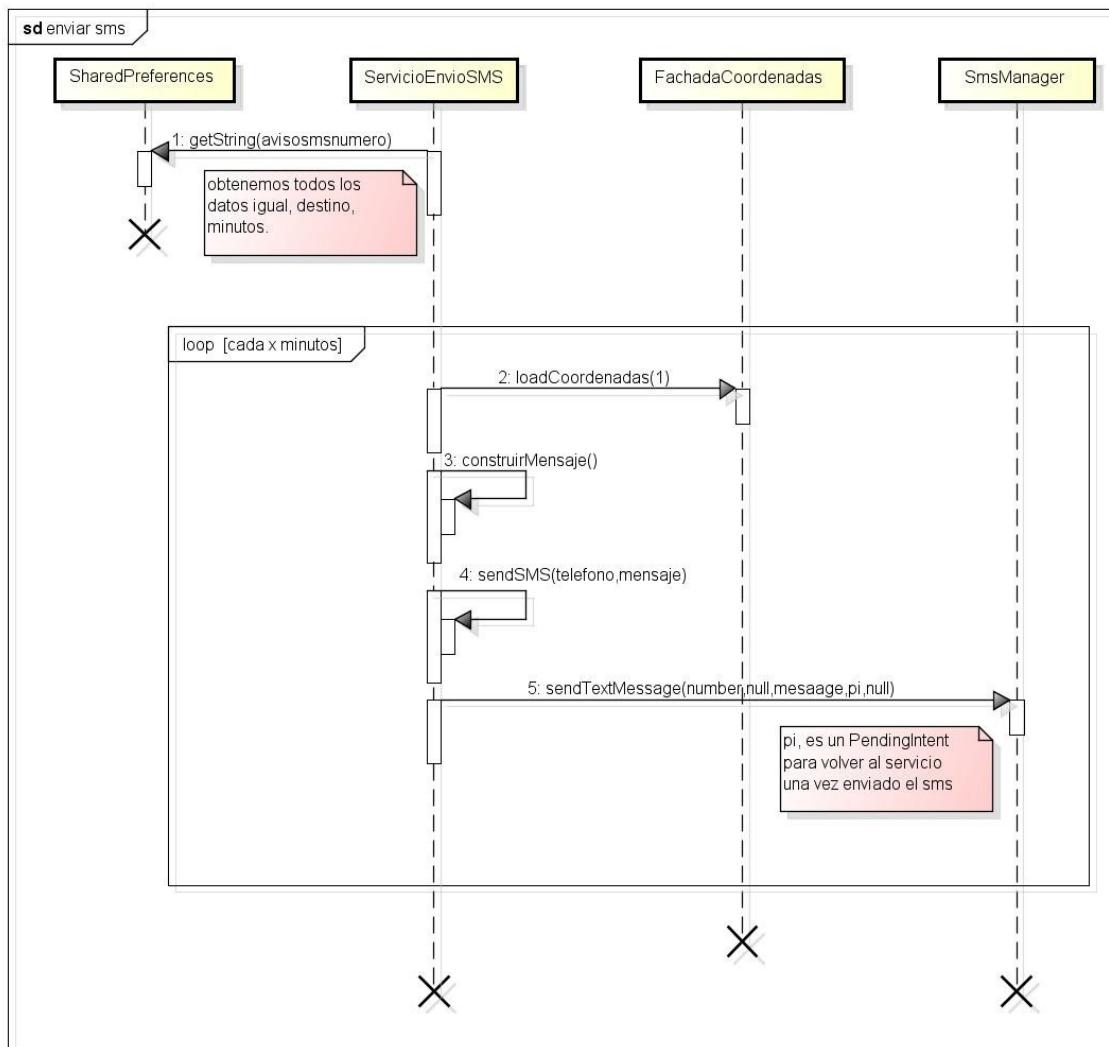


Ilustración 68: gps enviar sms

### 6.3.5 Autorranque al encender

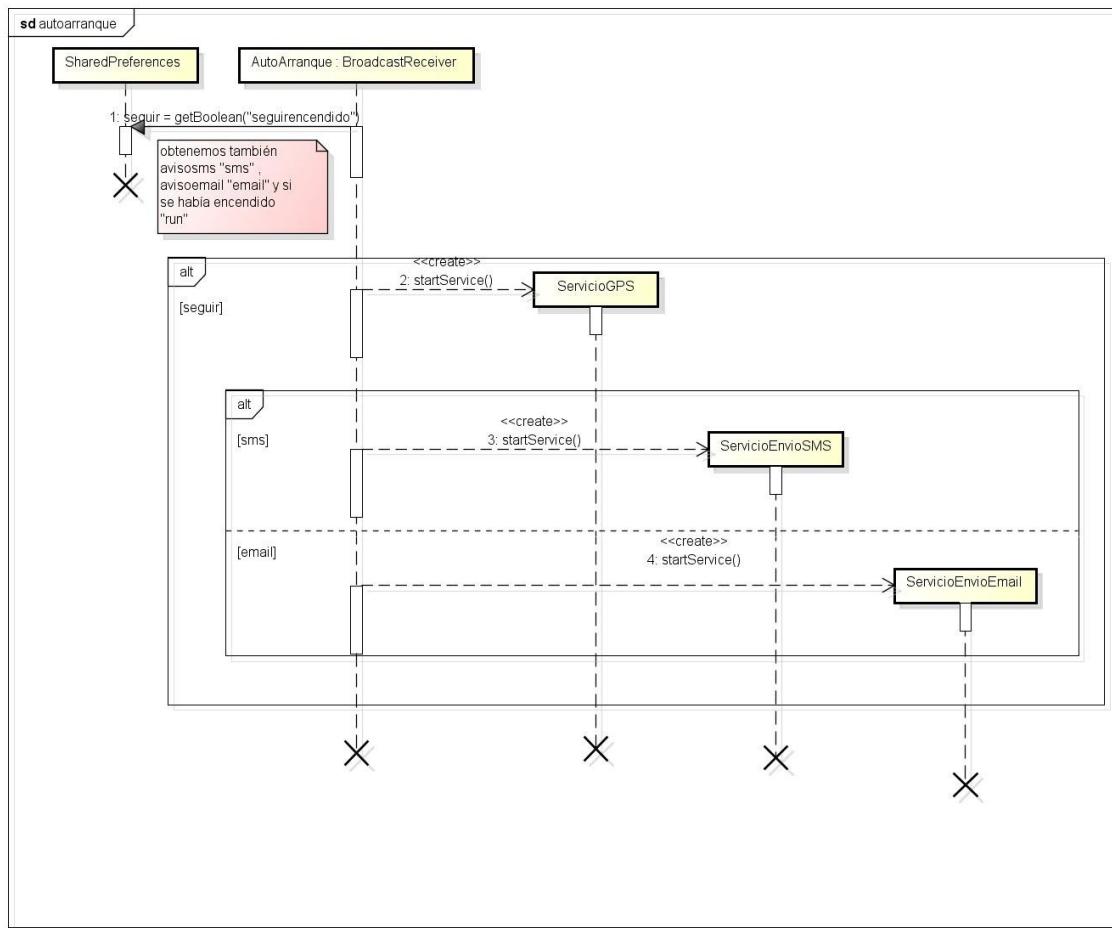


Ilustración 69: gps autoarranque

## 6.4 Módulo Consola de Acessos

### 6.4.1 Añadir servidor SSH

La forma de añadir un servidor ssh es la misma que para un servidor email, y ambas, se realizan de la misma forma que en el Módulo SSH, por lo que no se añadirá un nuevo diagrama.

### 6.4.2 Monitorizar servidor SSH

Monitorizar un servidor ssh y un servidor email es igual, por lo que solo se mostrará el diagrama de secuencia de uno de ellos, en este caso de ssh.

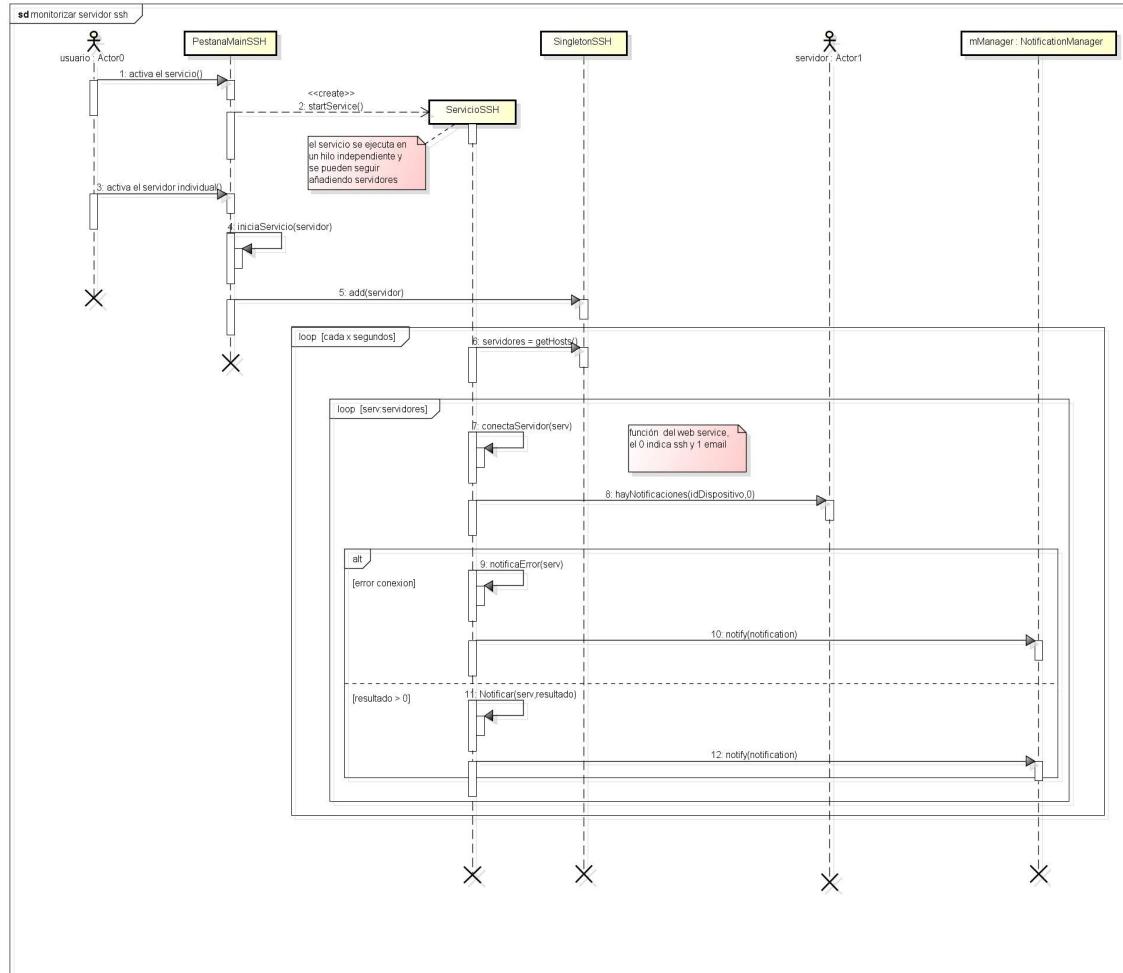


Ilustración 70: control monitorizar

#### 6.4.3 Descargar通知aciones nuevas y guardadas

El proceso para descargar通知aciones nuevas y guardadas es el mismo, solo cambiando el método al que se invocará en el Web Service, por lo que solo se mostrará un diagrama de secuencia, representando la acción de descargar las通知aciones guardadas.

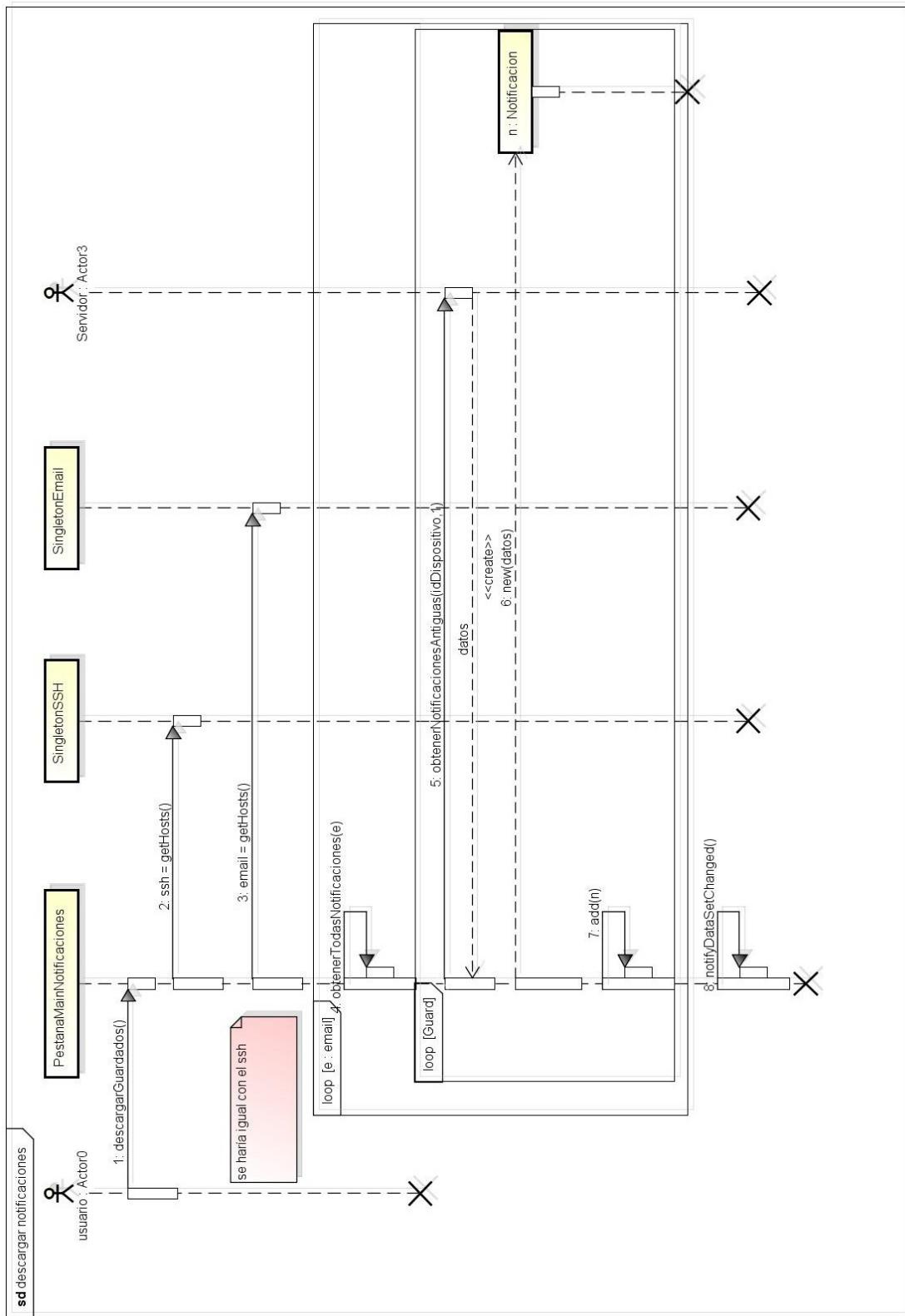


Ilustración 71: control descargar

#### **6.4.4 Filtrar notificaciones**

Para comprender los siguientes diagramas hay que tener en cuenta que las notificaciones que se muestran en la lista están guardadas en memoria en un `ArrayList<Notificacion>` llamado **datos** el cual es un atributo de `PestanaMainNotificaciones`.

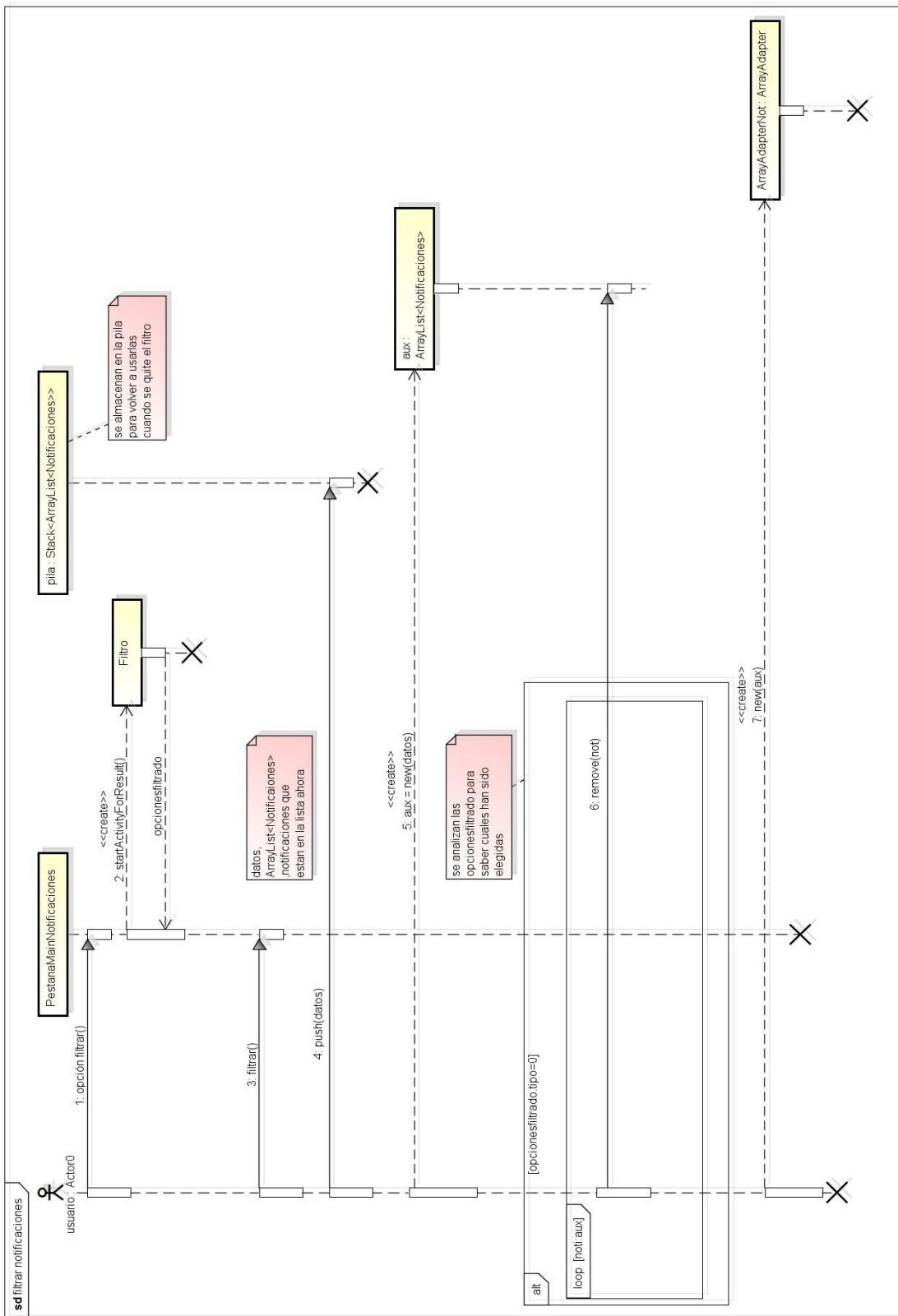


Ilustración 72. control filtrar

#### 6.4.5 Eliminar filtro

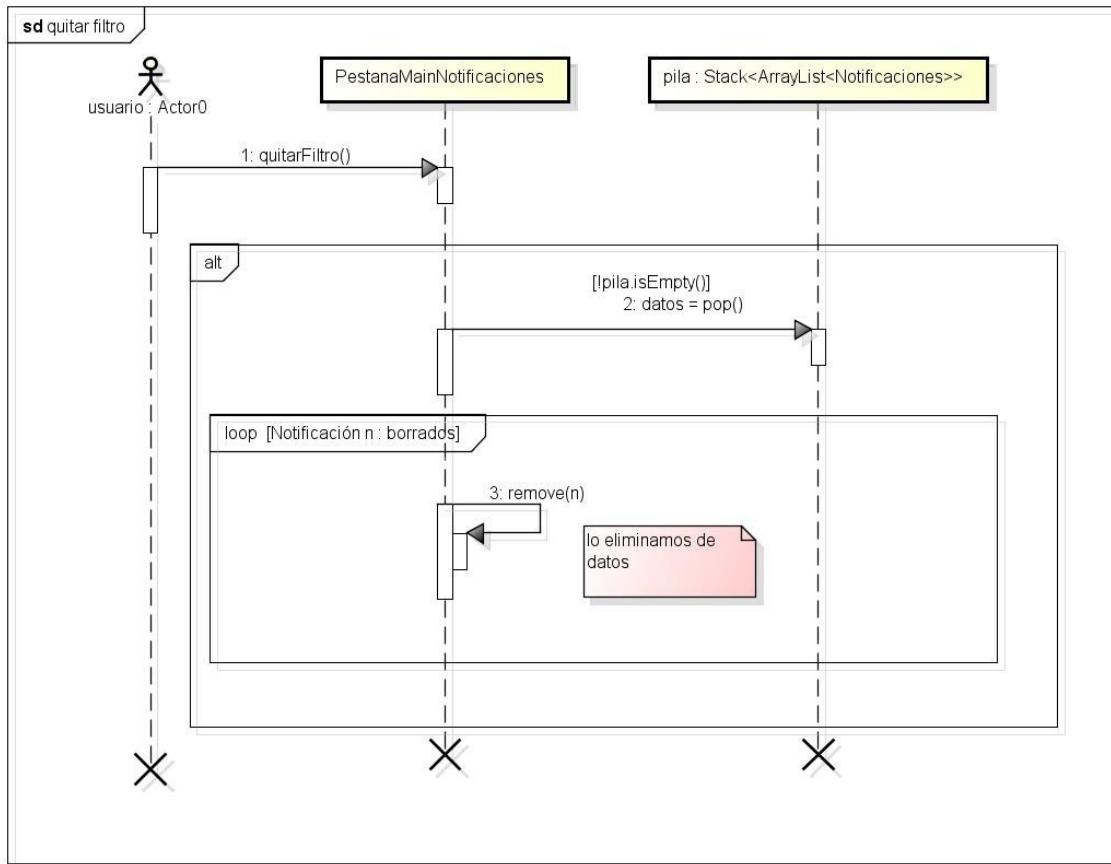


Ilustración 73: control quitar filtro

#### 6.4.6 Borrar notificaciones

Se puede borrar una notificación sola mediante el menú contextual que se abre al hacer una pulsación larga sobre el elemento de la lista, o borrar todas las notificaciones que están actualmente en la lista. El mecanismo en ambos casos es el mismo por lo que solo se mostrará el diagrama de secuencia de borrar todas las notificaciones de la lista.

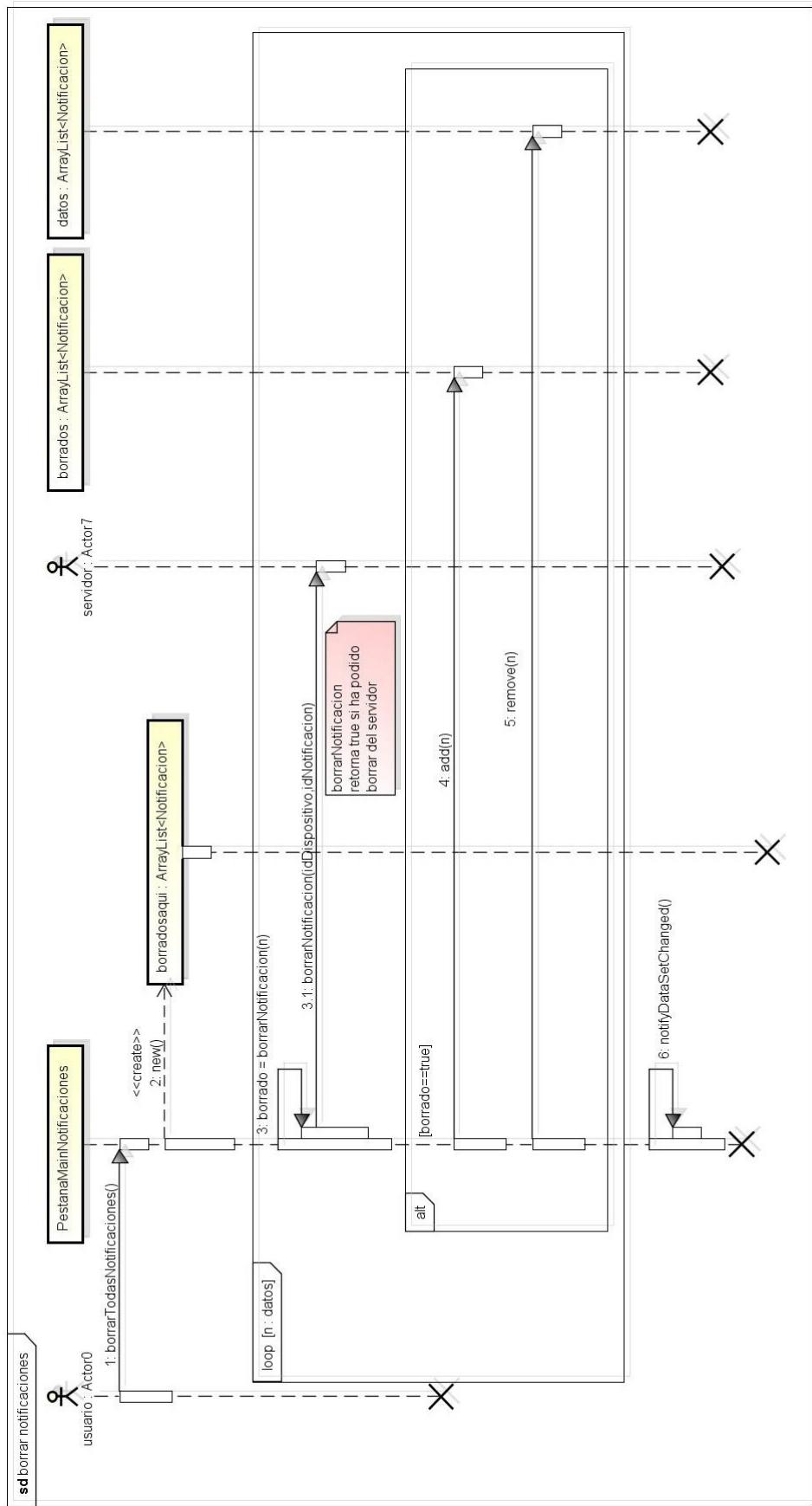


Ilustración 74: control borrar

## 6.5 Web Service

### 6.5.1 Borrar notificación

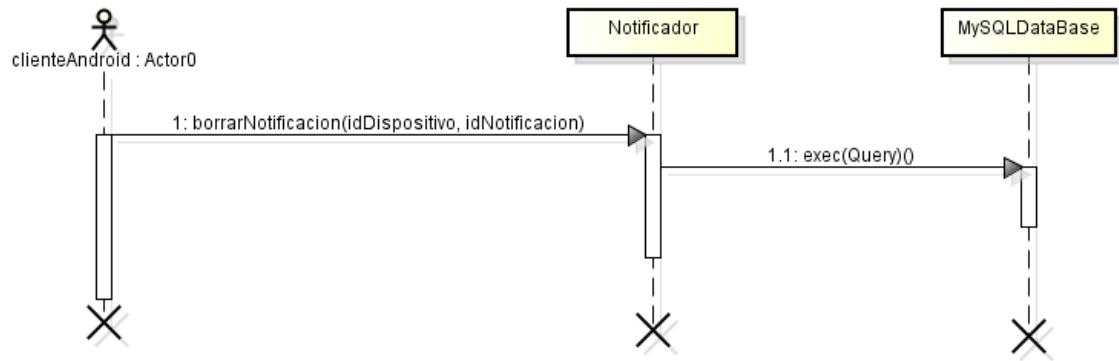


Ilustración 75: Borrar notificación

### 6.5.2 Hay notificaciones

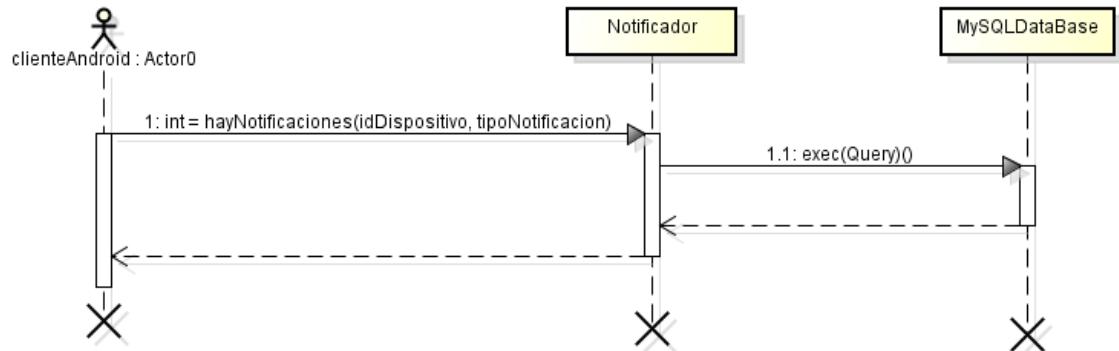


Ilustración 76: Hay notificaciones

### 6.5.3 Obtener notificaciones



Ilustración 77: Obtener notificaciones

#### 6.5.4 ScanLog

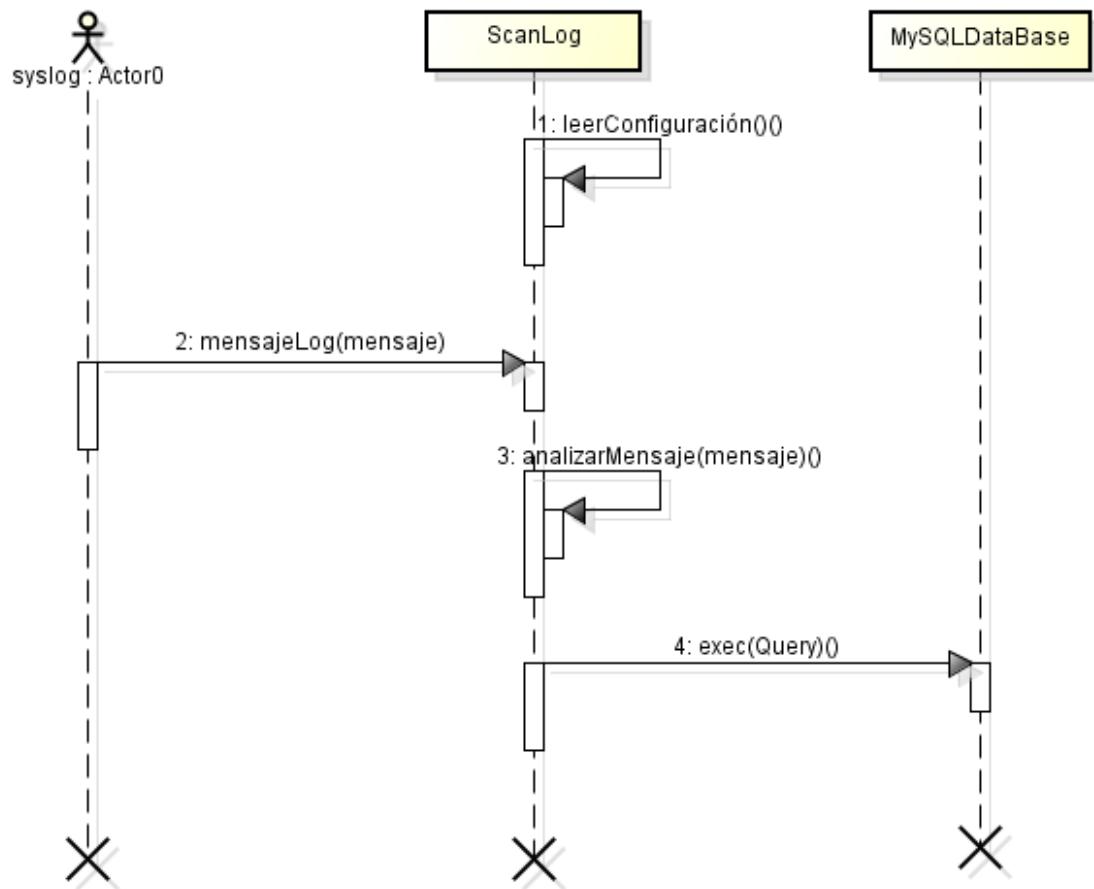


Ilustración 78: Scan Log

## 7 REFERENCIAS CRUZADAS A LOS REQUISITOS

En la siguiente tabla se muestran todos los requisitos funcionales y las clases que lo realizan.

RFXX	APLICACIÓN.PAQUETE.CLASE
RF01	WebService.ScanLog.ScanLog.java
RF02	WebService.ScanLog.ScanLog.java
RF03	WebService.Notificador.Notificador.java
RF04	WebService.Notificador.Notificador.java
RF05	WebService.Notificador.Notificador.java
RF06	ControlAccesos.vista.ServicioEmail/SSH.java
RF07	TerminalSSH.vista.Consola.java
RF08	TerminalSSH.vista.PestañaMainFav.java
RF09	TerminalSSH.vista.ComandosEjecutados.java
RF10	TerminalSSH.vista.ComandosEjecutados.java
RF11	LocalizadorGPS.vista.ServicioEnvioEmail/SMS.java
RF12	LocalizadorGPS.vista.ServicioGPS.java
RF13	LocalizadorGPS.vista.ServicioXXX.java
RF14	LocalizadorGPS.vista.LectorSMS.java
RF15	LocalizadorGPS.vista.LectorSMS.java
RF16	LocalizadorGPS.vista.LocalizadorGPSActivity.java

Tabla 41: Referencia a los requisitos

## 8 PRUEBAS

Al usar una metodología de desarrollo basada en Scrum , siempre se dispondrá de un aplicación en funcionamiento pero con escasa funcionalidad, a medida que va pasando el tiempo se aumenta la funcionalidad de la aplicación manteniendo la parte del ciclo anterior.

En cuanto se tiene una aplicación con una parte de funcionalidad se prueba en un ámbito simulado (servidores simulados , llamadas simuladas , etc) encontrando todos los fallos posibles y resolviéndolos en la siguiente iteración. Es por esto que la aplicación se irá depurando a medida que se avance en el desarrollo del proyecto.

### 8.1 Pruebas de stress

Son realizadas para verificar que el sistema funciona como se espera bajo grandes volúmenes de transacciones y peticiones.

Para ello se monitorizará el teléfono mientras se envían gran cantidad de eventos como clicks, gestos y pulsaciones táctiles.

Además, se utilizará el programa Jmeter para verificar que el Web Service puede atender a peticiones masivas sin colapsarse.

## 8.2 Pruebas de integración

Estas pruebas consisten en verificar que un gran conjunto de partes software que funcionan juntos lo hacen de forma correcta . Es la fase de test en la que los módulos individuales (clases) son combinados y probados como grupo.

Para realizar estas pruebas se ejecutará la aplicación mientras se monitoriza todo su sistema de log para comprobar que no se producen fallos , además se probará la aplicación en todos los ámbitos posibles (sin conexión a internet, sin base de datos , sin gps , etc).

## 8.3 Tablas de pruebas

En este apartado se incluirán tablas en las que se muestran pruebas que se realizan a la aplicación y su resultado esperado.

Como siempre se dividirá en los distintos módulos de los que consta la aplicación.

### 8.3.1 Principal

Código	Descripción	Salida esperada
P01	No hay fichero de preferencias.	Utiliza los valores por defecto
P02	No están las aplicaciones instaladas.	Se muestra un mensaje de error.

Tabla 42: Pruebas principal

### 8.3.2 Módulo SSH

#### 8.3.2.1 AccesoDatos

Código	Descripción	Salida esperada
SA01	No hay base de datos.	Se crea una nueva base de datos.
SA02	La base de datos está corrupta.	Error del sistema.
SA03	Se hace un “load” con la base de datos vacía.	Devuelve lista vacía.
SA04	Se intenta borrar un servidor/script que no existe.	No hacer nada.
SA05	Se intenta editar un servidor que no existe.	No hacer nada.
SA06	Se borra la tabla y se inserta un nuevo elemento.	Insercción correcta.
SA07	Se borra la tabla sin ningún elemento.	Se borra la tabla.
SA08	Se borra la tabla con elementos.	Se borra la tabla.

Tabla 43: Pruebas ssh accesodatos

### 8.3.2.2 Vista

Código	Descripción	Salida esperada
SV01	No hay fichero de preferencias.	Utiliza los valores por defecto
SV02	Usuario o contraseña son campos vacíos.	Se muestra un mensaje de error al conectarse.
SV03	El puerto al que conectarse no es un número.	No se hace nada.
SV04	No hay conexión a internet.	Se muestra un mensaje de error al conectarse.
SV05	El servidor al que se desea conectar está offline.	Se muestra un mensaje de error al conectarse.
SV06	Intenta guardar comandos sin haber ejecutado ninguno.	Se muestra mensaje de error.
SV07	Intenta reconectarse con una sesión abierta .	Se muestra mensaje de error.
SV08	Se guarda un script sin nombre.	Insercción correcta con nombre igual a la cadena vacía.
SV09	Se ejecuta logout/exit en la consola.	Error del sistema.
SV10	Se ejecuta un script sin comandos.	No hace nada.

Tabla 44: Pruebas ssh vista

### 8.3.3 Módulo GPS

#### 8.3.3.1 AccesoDatos

Las pruebas para este paquete son las mismas que para el paquete accesodatos del módulo SSH , ya que la funcionalidad es la misma.

### 8.3.3.2 Vista

Código	Descripción	Salida esperada
GV01	No hay fichero de preferencias.	Utiliza los valores por defecto
GV02	No hay gps activo.	No hace nada, pero no usa gps para localizar.
GV03	No hay red activa.	No se hace nada, pero no se usa la red para localizar.
GV04	No hay conexión a internet.	No hace nada.
GV05	No se consiguen obtener localizaciones.	No se hace nada.
GV06	Se intenta mostrar una localización sin datos.	Se muestra mensaje de error.
GV07	No hay coordenadas en la bd.	Se muestra mensaje de error o no se envía nada.
GV08	Datos incorrectos para enviar el email.	No se hace nada.
GV09	Datos incorrectos para enviar el sms.	No se hace nada.

Tabla 45: Pruebas gps vista

### 8.3.4 Módulo Control de Accesos

#### 8.3.4.1 Vista

Código	Descripción	Salida esperada
AV01	No hay fichero de preferencias.	Utiliza los valores por defecto
AV02	El puerto no es un campo numérico.	Mensaje de error al insertar.
AV03	No hay servidores a los que monitorizar.	No se hace nada.
AV04	No hay conexión a internet.	Mensaje de error.
AV05	Servidor al que se quiere monitorizar está offline.	Mensaje de error.
AV06	Datos del servidor incorrectos.	Mensaje de error.
AV07	Se borra una notificación de un servidor al que no se puede acceder.	No se borra nada.
AV08	Se descargan notificaciones de un servidor que no tiene.	No se descarga nada.
AV09	Se filtra con un criterio que ninguna notificación cumple.	Lista vacía.
AV10	Se intenta descargar notificaciones con un filtro aplicado.	Mensaje de error y no se descarga nada.

Tabla 46: Pruebas accesodatos vista

### 8.3.5 Web Service

Código	Descripción	Salida esperada
WS01	No hay fichero de preferencias.	No hace nada.*
WS02	No hay acceso a la base de datos.	Mensaje de error(Retorna -1)
WS03	Recibe una petición de un nuevo dispositivo.	Lo añade a la BD.
WS04	No se conecta al syslog.	No hace nada.

Tabla 47: Pruebas webservice

\* Como no hay datos de configuración no será posible conectarse a la base de datos, ni leer del syslog, por lo que aparecerán los mismos errores que en WS02 y WS04.

## 9 ENTORNO TECNOLÓGICO DEL SISTEMA

En este apartado se describe la parte física del equipo, el equipo lógico y las comunicaciones que existen entre todo el sistema, además de las restricciones técnicas que existen.

## 9.1 Restricciones técnicas

El proyecto va a ser desarrollado por una parte para Android, y otra parte para un servidor Linux.

Para poder aprovechar todo el potencial de la aplicación será necesario disponer de un terminal Android 2.3.3 o superior y de un servidor Linux, preferiblemente con sistema operativo Ubuntu Server 10.04, Java JRE 6.0, Apache Tomcat 7.0, Apache Axis2 1.6.2 y MySQL 5.0.

Mediante el siguiente diagrama se mostrarán los distintos componentes necesarios para el funcionamiento correcto de la aplicación y las relaciones existentes entre ellos de forma simplificada.

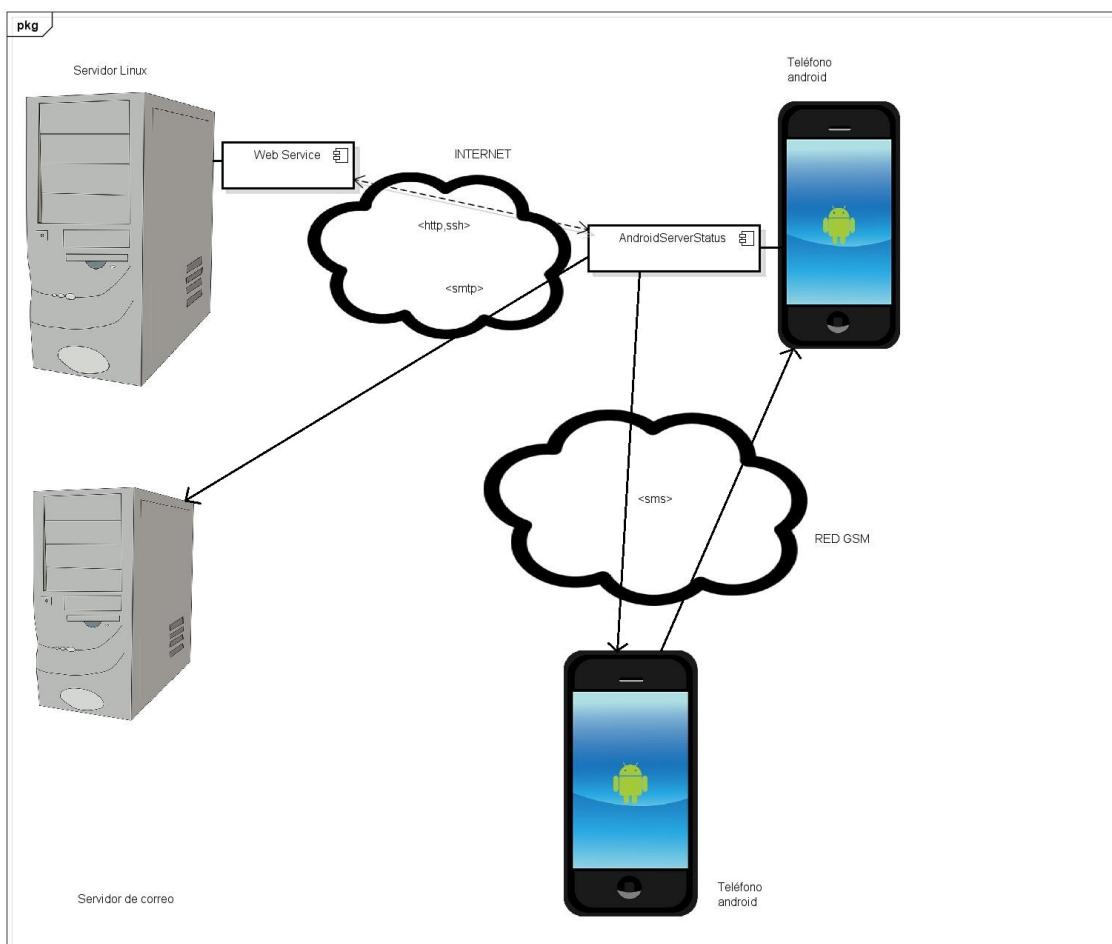


Ilustración 79: Diagrama componentes

En el diagrama se muestra como, el dispositivo “Android 1”, el cual tiene instalada nuestra aplicación, es capaz de comunicarse con el servidor a través de Internet, tanto para utilizar la consola ssh como para recibir las notificaciones desde el Web Service.

Además también utilizará la conexión a Internet para enviar emails con las coordenadas obtenidas recientemente.

Por otro lado, y utilizando otro medio de comunicación como es la red GSM

(llamadas y mensajes de texto) los dos dispositivos móviles con S.O Android se podrán comunicar, tanto para activar el servicio de localización como para enviar la última coordenada GPS obtenida.

## 9.2 Comunicación dentro del sistema

A continuación se mostrará de forma más detallada las distintas comunicaciones que se podrán dar en el sistema. La más importante es la que se dará entre el dispositivo Android y el servidor Linux.

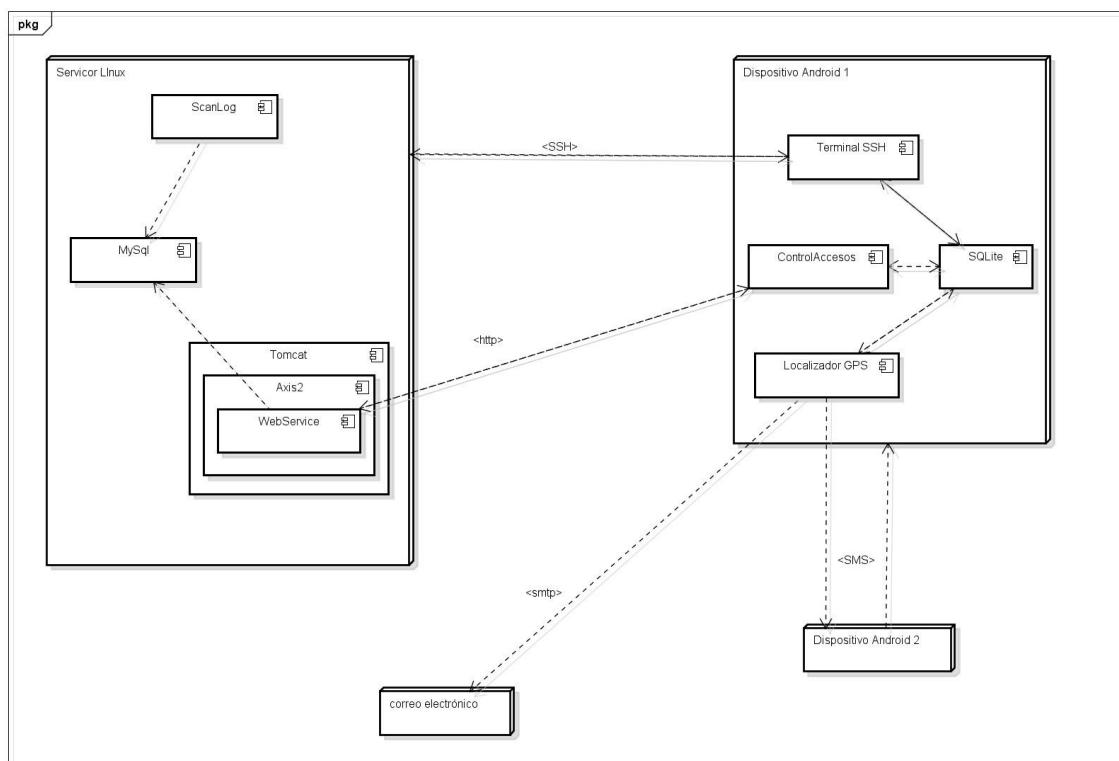


Ilustración 80: Diagrama componentes completo

En este caso el “Dispositivo Android 1” tiene instalada nuestra aplicación, y el “Servidor Linux” tiene instalado el Web Service junto con el resto de componentes que necesita.

Se ve como se realiza la comunicación entre los distintos módulos, entre dispositivo Android y el servidor mediante Internet, y entre dispositivos Android a través de sms.



# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

**Anexo IV - Documentación técnica de programación**

**Alumnos**      David Herrero de la Peña  
                  Jonatan Santos Barrios

**Tutor**      Carlos López Nozal  
                  DEPARTAMENTO DE INGENIERÍA CIVIL  
                  Área de Lenguajes y Sistemas Informáticos  
                  Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## Índice de contenido

1. Introducción.....	172
2. Documentación de las bibliotecas.....	172
2.1. MySQLJDBC.....	172
2.2. Jsvc.....	172
2.3. Jsch.....	172
3. Manual del programador.....	172
3.1. Instalación Java JDK.....	173
3.1.1. Instalación en Windows.....	173
3.1.2. Instalación en Linux (Ubuntu).....	173
3.2. Instalación Android SDK.....	174
3.3. Instalación Eclipse.....	175
3.3.1. Instalación plugin ADT.....	176
3.3.2. Instalación librerías externas.....	176
3.4. Instalación Apache Tomcat 7.0.....	177
3.5. Instalación Apache Axis2.....	179
3.6. Instalación MySQL.....	180
3.6.1. Instalación conector MySQLJDBC.....	181
3.7. Configuración Syslog.....	181
3.8. Instalación Jsvc.....	182
4. Manual de compilación.....	182
4.1. Compilación de aplicaciones para Android.....	182
4.2. Compilación del Web Service.....	184
4.3. Compilación del Demonio.....	193
5. Pruebas.....	197
5.1. Pruebas de integración.....	197
5.1.1. Pruebas sobre el Launcher.....	197
5.1.2. Pruebas sobre el Terminal SSH.....	197
5.1.3. Pruebas sobre el Control de Accesos.....	198
5.1.4. Pruebas sobre el localizador GPS.....	199
5.2. Pruebas de stress.....	199

## Índice de ilustraciones

Ilustración 81: Página de descarga de Java JDK.....	173
Ilustración 82: Página de descarga de Android SDK.....	174
Ilustración 83: Instalación de plataformas de Android.....	175
Ilustración 84: Página de descarga de Eclipse.....	175
Ilustración 85: Instalación Plugin ADT.....	176
Ilustración 86: Instalación librerías externas.....	177
Ilustración 87: Página principal de Apache Tomcat 7.0.....	179
Ilustración 88: Página principal de Apache Axis2.....	180
Ilustración 89: Iniciar ADV Manager.....	182
Ilustración 90: Configuración dispositivo virtual Android.....	183
Ilustración 91: Compilación aplicación Android.....	183
Ilustración 92: Menú desplegable para creación de contenedor de Web Services.....	184
Ilustración 93: Selección de Web Service.....	185
Ilustración 94: Elección Web Service Runtime.....	186
Ilustración 95: Configuración Web Service.....	187
Ilustración 96: Iniciar servidor.....	188
Ilustración 97: Menú desplegable compilación Web Service.....	189
Ilustración 98: Selección de servidor donde correr el Web Service.....	190
Ilustración 99: Añadir la clase del Web Service.....	191
Ilustración 100: Pantalla principal Apache Axis2.....	192
Ilustración 101: Listado Web Services publicados.....	192
Ilustración 102: Creación archivo aar para Web Service.....	193
Ilustración 103: Menú desplegable para exportación.....	194
Ilustración 104: Selección de exportación a archivo JAR.....	195
Ilustración 105: Datos de exportación del demonio.....	196
Ilustración 106: Creación de grupo de hilos.....	199
Ilustración 107: Configuración del numero de hilos.....	200
Ilustración 108: Añadir muestreador Web Service.....	200
Ilustración 109: Configuración datos Web Service.....	201
Ilustración 110: Añadir Gráfico de Resultados.....	201
Ilustración 111: Resultado de las pruebas de stress.....	202

## Índice de tablas

Tabla 48: Pruebas sobre Launcher.....	197
Tabla 49: Pruebas sobre Terminal SSH.....	197
Tabla 50: Pruebas sobre Control de accesos.....	198
Tabla 51: Pruebas sobre Localizador GPS.....	199

## **LISTA DE CAMBIOS**

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>
1.0	15/05/12	Primera versión: Introducción.
2.0	10/06/12	Segunda versión: Documentación de las bibliotecas, Manual del programador.
3.0	18/06/12	Tercera versión: Manual de compilación.
4.0	21/06/12	Cuarta versión: Pruebas.

## 1 INTRODUCCIÓN

Este anexo recoge toda la documentación técnica necesaria sobre como se ha implementado la aplicación, indicando los pasos necesarios para generar la aplicación partiendo del código fuente, resaltar que el desarrollador deberá tener conocimientos de administración de servidores para poder comprender algunos de los pasos seguidos.

Para ello dispone de información sobre que bibliotecas han sido utilizadas, partes en necesario y los pasos a realizar para compilar el código, junto con las pruebas.

## 2 DOCUMENTACIÓN DE LAS BIBLIOTECAS

En este apartado se aporta información sobre que bibliotecas se han requerido para la realización de la aplicación.

### 2.1 MySQLJDBC

Los datos de las notificaciones están almacenados en una BD de tipo MySQL en el equipo servidor, para poder acceder a ellos, es necesario el uso de un conector que proporcione la conexión al fichero de la BD y permita realizar las consultas necesarias. Para ello se usa el conector MySQLJDBC [1], que permite realizar cualquier tipo de consulta sobre la BD.

### 2.2 Jsvc

Para mantener activa la escucha de los logs en nuestra aplicación, es necesario que esta se ejecute con demonio o servicio. Jsvc [2] son un conjunto de librerías, que permiten que las aplicaciones Java se ejecuten en UNIX con más facilidad. Estas librerías son imprescindibles para la creación del demonio correspondiente a partir de nuestra aplicación Java.

### 2.3 Jsch

A la hora de mantener una conexión entre el dispositivo móvil y el servidor, se pretende tener una cierta seguridad en la transferencia de información, por ello se utiliza una conexión ssh. Para poder realizar la conexión se utilizó Jsch [3] que es una librería de utilidades de comunicación para Java, principalmente es usada para conexiones de tipo ssh1 y ssh2.

## 3 MANUAL DEL PROGRAMADOR

Este apartado pretende ser una guía de referencia para futuros programadores de la aplicación, facilitándoles en la medida de lo posible, la ampliación de las funcionalidades de la aplicación.

### 3.1 Instalación Java JDK

El primer requisito necesario para poder compilar la aplicación es disponer del entorno de desarrollo de Java instalado en el equipo.

Para ello lo primero que se debe realizar es la descarga de los archivos de instalación desde la página web oficial [4], donde se escogerá la opción Java Platform (JDK), a continuación se elige la arquitectura deseada y se descarga.

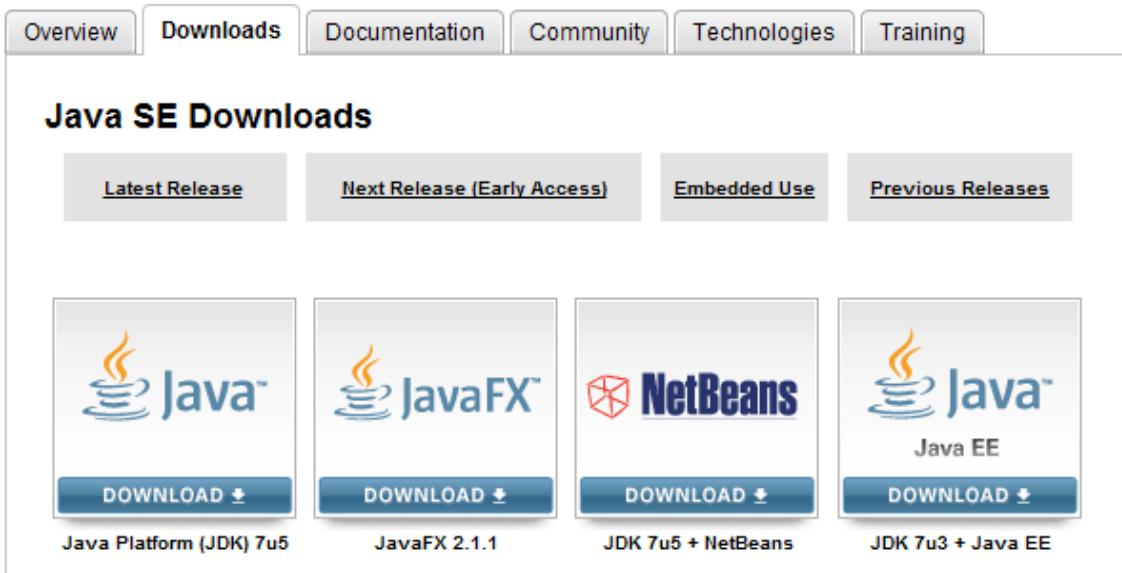


Ilustración 81: Página de descarga de Java JDK

#### 3.1.1 Instalación en Windows

Se ejecuta el asistente paso a paso aceptando cada uno de ellos hasta completar la instalación.

#### 3.1.2 Instalación en Linux (Ubuntu)

Una vez descargado el archivo bin de la página web, hay que cambiarle los permisos al fichero para que se pueda ejecutar e instalarlo. Para ello desde el terminal ejecutar:

```
$ cd /ruta de descarga del fichero
$ chmod +x jdk-6-linux-i586.bin
$ sudo ./jdk-6-linux-i586.bin
```

A partir de aquí se iniciara la instalación una vez finalizada se debe mover la carpeta creada.

```
$ sudo mv jdk1.6.0 /usr/lib/jvm
```

El siguiente paso es configurar la nueva instalación como una de las “alternativas de Java”:

```
$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/lib/jvm/jdk1.6.0/bin/java" 1
```

Y por ultimo hay que configurar la “nueva alternativa” como la usada por defecto:

```
$ sudo update-alternatives --set java /usr/lib/jvm/jdk1.6.0/bin/java
```

### 3.2 Instalación Android SDK

Para poder usar, compilar y ejecutar la aplicación es necesario instalar el SDK de Android para la plataforma software a emplear [5].

#### Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*.

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r18-windows.zip</a>	37448775 bytes	bfbfdf8b2d0fdecc2a621544d706fa98
	<a href="#">installer_r18-windows.exe</a> (Recommended)	37456234 bytes	48b1fe7b431afe6b9c8a992bf75dd898
Mac OS X (intel)	<a href="#">android-sdk_r18-macosx.zip</a>	33903758 bytes	8328e8a5531c9d6f6f1a0261cb97af36
Linux (i386)	<a href="#">android-sdk_r18-linux.tgz</a>	29731463 bytes	6cd716d0e04624b865ffed3c25b3485c

Ilustración 82: Página de descarga de Android SDK

Para la instalación solo es necesario ejecutar el asistente de instalación paso a paso hasta finalizar con la instalación.

Una vez instalado y ejecutado lo primero es seleccionar las plataformas de Android para las que queremos desarrollar y aceptar la instalación.

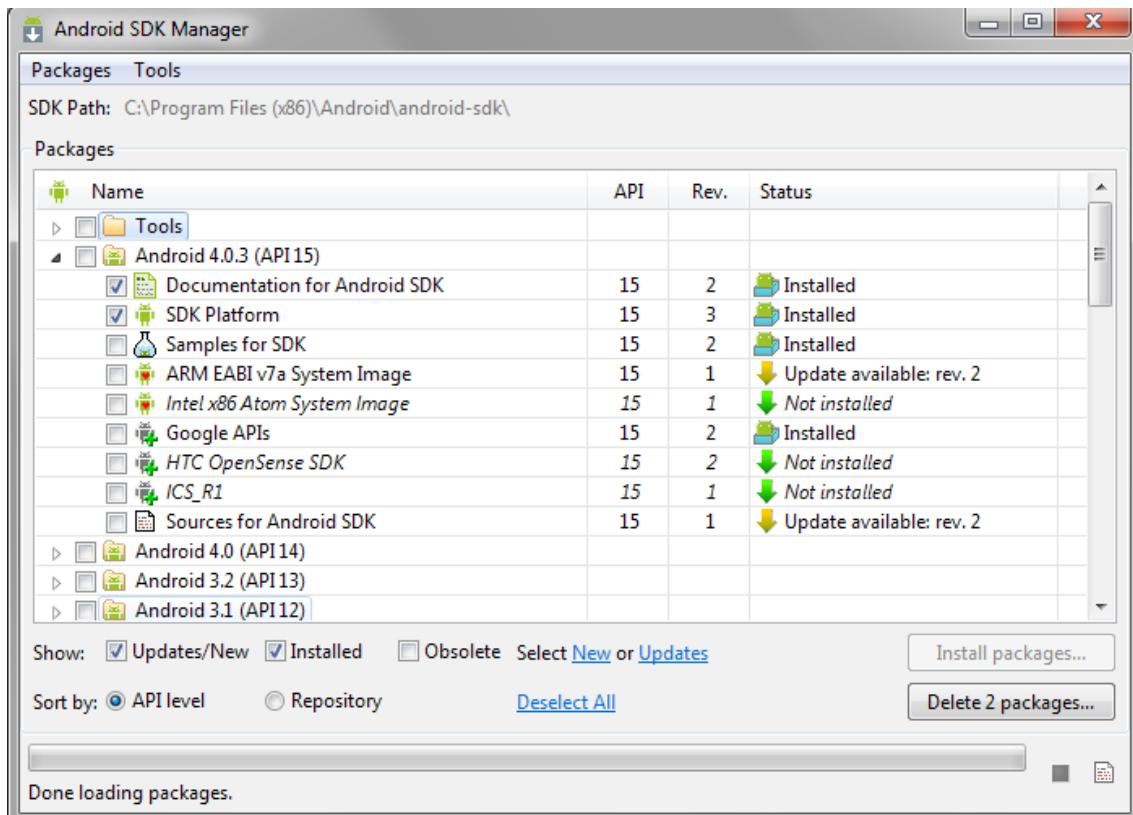


Ilustración 83: Instalación de plataformas de Android

### 3.3 Instalación Eclipse

Una vez instalado el entorno de ejecución de Java y Android SDK en el equipo, se descargará e instalará Eclipse [6]. La versión de Eclipse que se debe instalar es Eclipse IDE for Java EE Developers, pues esta versión nos permite la compilación de Web Service.



Ilustración 84: Página de descarga de Eclipse

Para realizar la instalación simplemente se descomprime el archivo descargado en la carpeta donde se deseé, y listo.

### 3.3.1 Instalación plugin ADT

Para poder trabajar en Eclipse con el Android SDK, es necesario la instalación del plugin denominado Plugin ADT. Para instalarlo se iniciará el Eclipse, una vez arrancado, se selecciona Help > Install New Software, donde pone Work With, se introduce <https://dl-ssl.google.com/android/eclipse/> y pulsar intro.

Aparecerá Developer Tools, se selecciona y ya se puede avanzar con la instalación pulsando Next.

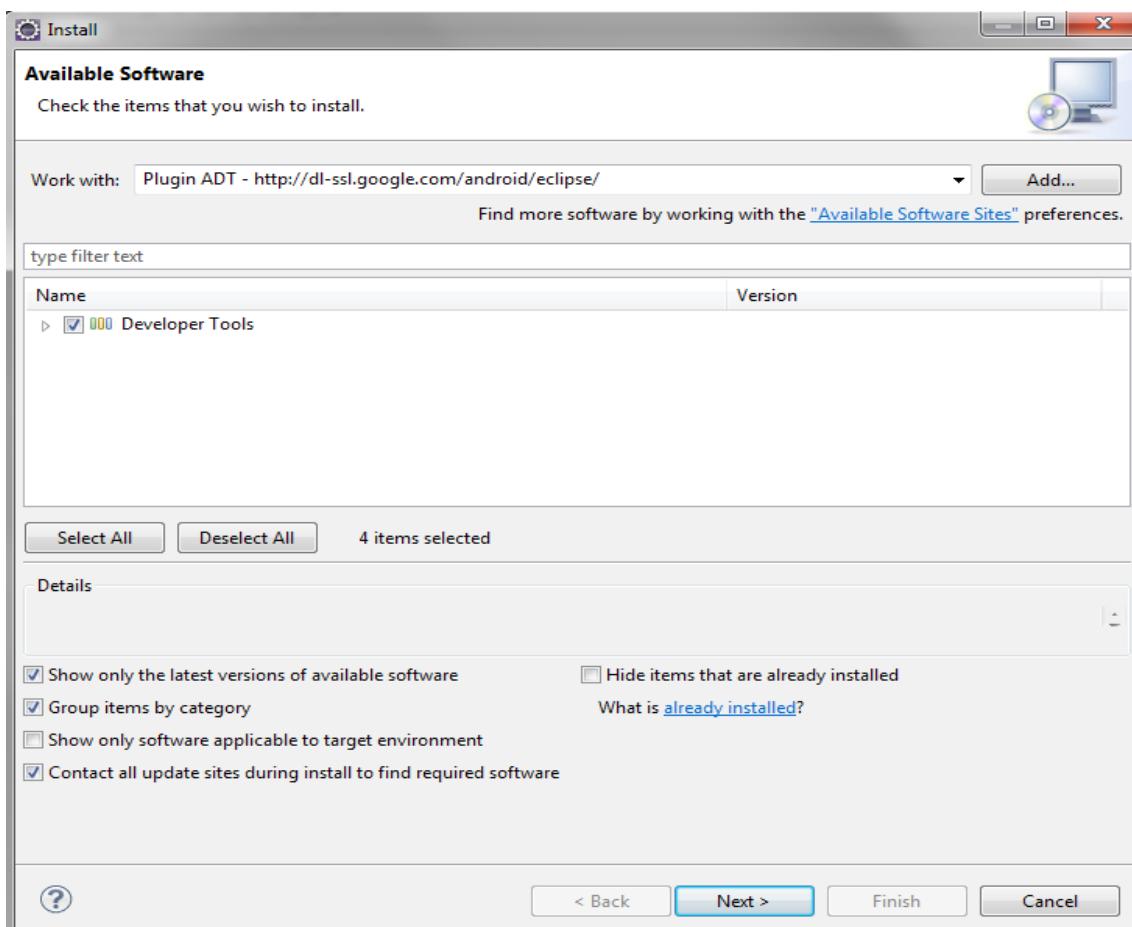


Ilustración 85: Instalación Plugin ADT

Se aceptaran todas las licencias de los plugins a instalar y un vez finalizada la instalación se reinicia Eclipse, después se selecciona Windows > Preferences > Android. Se pulsa en Browse y se introduce la dirección de la carpeta donde se encuentra instalado Android SDK y aceptar con el botón OK.

### 3.3.2 Instalación librerías externas

Para la compilación de algunas de las aplicaciones es necesario el uso de librerías externas, para poder hacer uso de ellas se debe añadir las librerías necesarias al proyecto de Eclipse, dentro del Eclipse se pulsa con el botón derecho sobre el proyecto y selecciona Properties, se pulsa sobre Java Build Path y sobre la pestaña Libraries pulsar

el botón Add External JARs, y seleccionar el archivo que contiene las librerías que se quieren añadir.

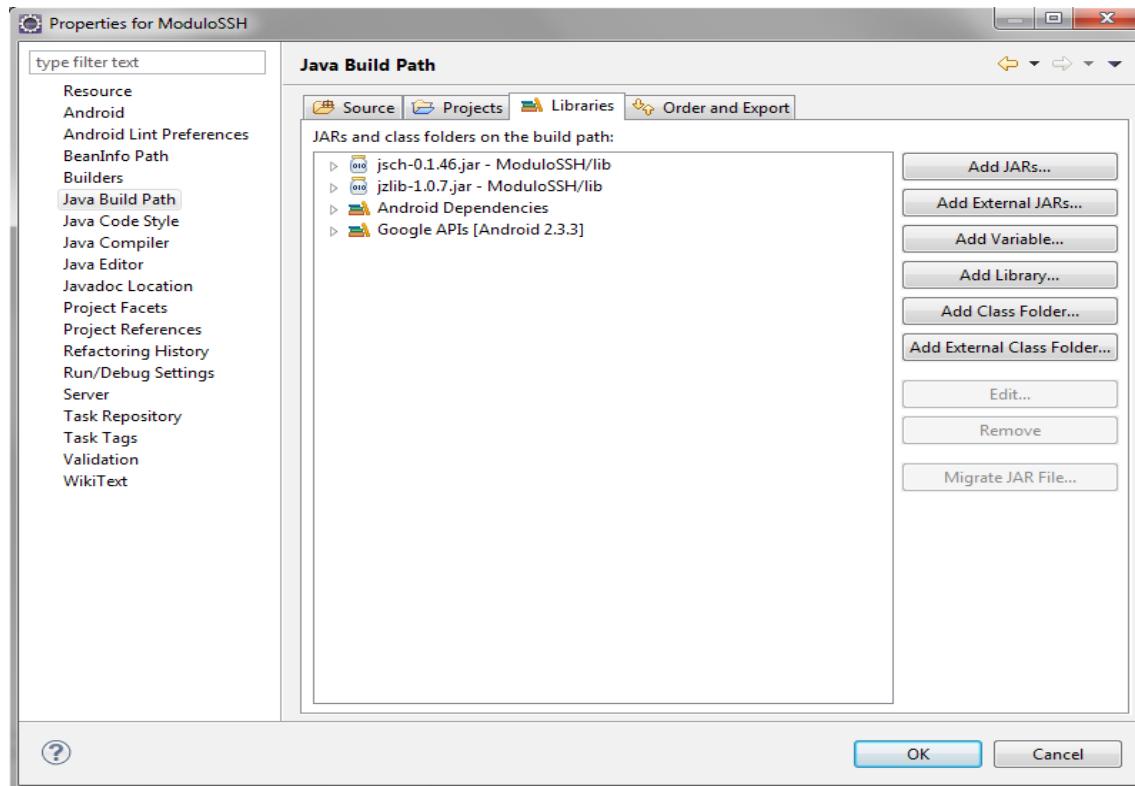


Ilustración 86: Instalación librerías externas

### 3.4 Instalación Apache Tomcat 7.0

Para la publicación del Web Service en la web, se necesita disponer de un servidor de aplicaciones o servlets, es por ello que se necesita la instalación de Apache Tomcat

El primer paso será descargar el archivo apache-tomcat-7.0.27.tar.gz de la página web de Tomcat [7].

El siguiente paso es descomprimir el archivo que se ha descargado anteriormente, usando el siguiente comando.

```
$ tar -xvzf apache-tomcat-7.0.27.tar.gz
```

Seguidamente se mueve la carpeta descomprimida a /usr/share/

```
$ sudo mv /home/alumno/apache-tomcat-7.0.27/ /usr/share/
```

Es necesario tener creada la variable de entorno JAVA\_HOME o JRE\_HOME para que funcione Tomcat, para ello editar el fichero catalina.sh.

```
$ sudo gedit /usr/share/apache-tomcat-7.0.27/bin/catalina.sh
```

Y agregar las siguientes líneas, justo después de la primera linea:

```
JAVA_HOME="/usr/lib/jvm/jdk1.7.0"  
JRE_HOME="/usr/lib/jvm/jdk1.7.0/jre"
```

Tener en cuenta que las rutas son de ejemplo, se deben introducir las rutas donde se tengan instalados los componentes.

Para levantar Tomcat se introducirá el siguiente comando:

```
$ sudo /usr/share/apache-tomcat-7.0.27/bin/startup.sh
```

Se comprobará que funciona correctamente ejecutando desde un navegador, la siguiente dirección:

<http://127.0.0.1:8080/>

Si la llamada se realiza desde un equipo externo cambiar por 127.0.0.1 por la ip del equipo servidor, y tener en cuenta que el firewall permita la entrada por ese puerto y que los equipo se encuentre en red.

Si se desea cambiar el puerto 8080 por otro hay que editar el fichero server.xml.

```
$ sudo gedit /usr/share/apache-tomcat-7.0.27/conf/server.xml
```

Y en la parte de fichero donde aparece:

```
<Connector port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />
```

Cambiar el puerto 8080 por el deseado. A continuación se deberá reiniciar el servidor con los siguientes comandos.

```
$ sudo /usr/share/apache-tomcat-7.0.27/bin/shutdown.sh
```

```
$ sudo /usr/share/apache-tomcat-7.0.27/bin/startup.sh
```

Y comprobar que corre en el puerto que hemos configurado, deberá aparecer la siguiente página.

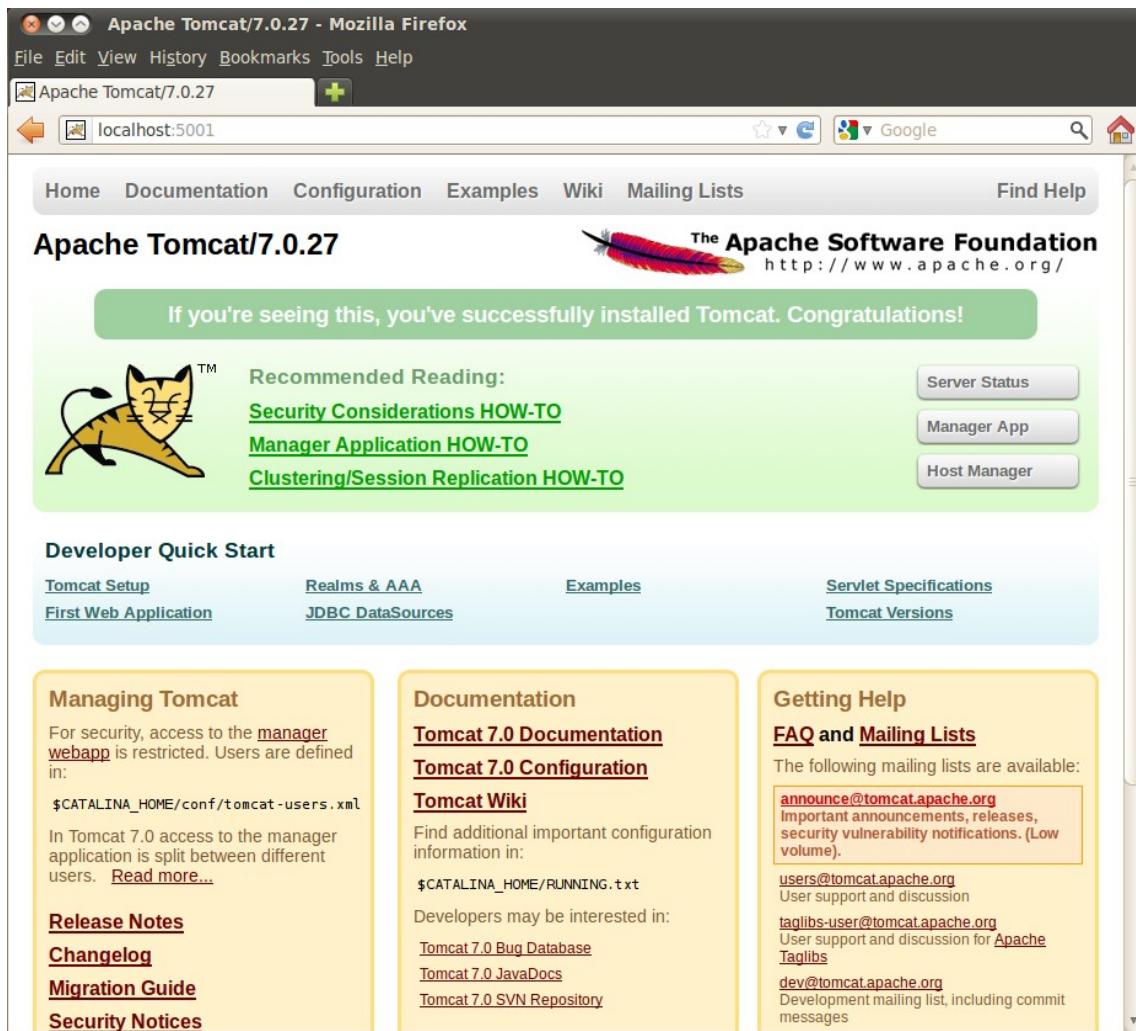


Ilustración 87: Página principal de Apache Tomcat 7.0

### 3.5 Instalación Apache Axis2

Para que el Web Service pueda correr y ser visible desde otros equipos, se debe introducir en un contenedor de servicios web, para ello se instalará Apache Axis2.

En primer lugar se descarga el fichero de distribución WAR de la página web [8].

Se descomprime el archivo, ya que vendrá en un archivo zip.

```
$ unzip axis2-1.6.2-war.zip
```

Y se mueve el archivo war a la carpeta /webapps/ de Tomcat.

```
$ sudo mv /home/alumno/axis2.war /usr/share/apache-tomcat-7.0.27/webapps/
```

Tomcat tardara unos pocos segundos en reconocerlo y lo instalará automáticamente, para comprobar que se instaló correctamente desde un navegador ir a la dirección correspondiente, tener en cuenta si se cambio el puerto por el que trabaja Tomcat.

<http://127.0.0.1:8080/axis2>

Deberá aparecer la página principal de Axis2.

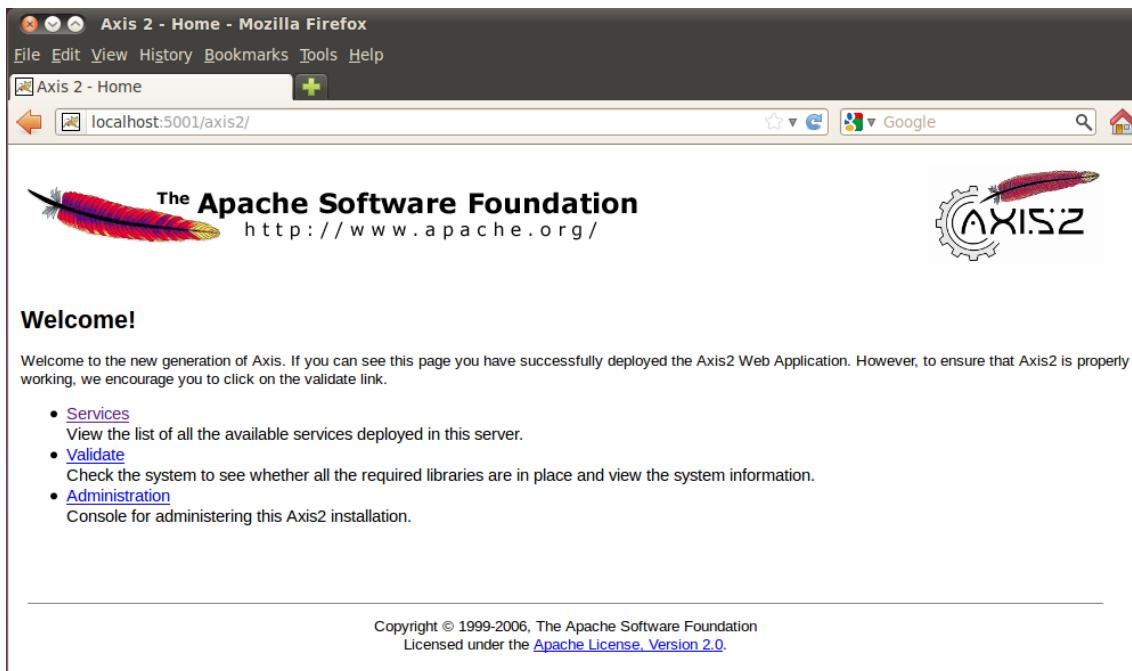


Ilustración 88: Página principal de Apache Axis2

### 3.6 Instalación MySQL

Para que el Web Service almacene los datos el equipo debe contar con una instalación de MySQL, un gestor de base de datos, para ello ejecutar el comando de instalación.

```
$ sudo aptitude install mysql-server
```

Y seguir los pasos indicados, a partir de aquí, crear los usuarios que se crean convenientes y con los permisos deseados.

La base de datos y las tablas correspondientes se crearán ejecutando las siguientes consultas.

```
CREATE DATABASE android_server_status;
```

```
CREATE TABLE IF NOT EXISTS `Dispositivos` (
`idDispositivo` int NOT NULL AUTO_INCREMENT,
`dispositivo` char(16) NOT NULL,
PRIMARY KEY (`idDispositivo`));
```

```

CREATE TABLE IF NOT EXISTS `Notificaciones` (
  `idNotificacion` int NOT NULL AUTO_INCREMENT,
  `tipoMensaje` int NOT NULL,
  `urgencia` int NOT NULL,
  `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `mensaje` longtext,
  PRIMARY KEY (`idNotificacion`));

```

```

CREATE TABLE IF NOT EXISTS `DispositivosNotificaciones` (
  `idDispositivo` int NOT NULL,
  `idNotificacion` int NOT NULL,
  `descargada` tinyint NOT NULL DEFAULT '0',
  PRIMARY KEY (`idDispositivo`, `idNotificacion`),
  FOREIGN KEY (`idDispositivo`) REFERENCES
  `Dispositivos`(`idDispositivo`)
  ON DELETE CASCADE,
  FOREIGN KEY (`idNotificacion`) REFERENCES
  `Notificaciones`(`idNotificacion`)
  ON DELETE CASCADE);

```

### 3.6.1 Instalación conector MySQLJDBC

Para poder acceder a la base de datos desde la aplicación Java se debe instalar un conector entre ambos, para ello se descargará el conector java de la página de MySQL [1]. Se descargará un archivo zip que se descomprime.

```
$ unzip mysql-connector-java-5.1.20.zip
```

Y se añade el conector a las librerías externas de java.

```
$ sudo mv /home/alumno/mysql-connector-java-5.1.20/mysql-connector-java-5.1.20-bin.jar /usr/lib/jvm/jdk1.7.0/jre/lib/ext/
```

### 3.7 Configuración Syslog

El demonio necesita recibir los mensajes de log desde la aplicación que registra y controla los logs, para ello se tiene que editar el archivo de configuración del servicio que realiza dicha tarea.

```
$ sudo gedit /etc/rsyslog.d/50-default.conf
```

El fichero final tendrá un aspecto como el que se muestra a continuación, las líneas añadidas son las resaltadas en negrita, donde se indica el puerto al que se quiere reenviar los mensajes log.

```

#
# First some standard log files. Log by facility.
#
auth,authpriv.*          @@localhost:5150
auth,authpriv.*           /var/log/auth.log
*.*;auth,authpriv.none   -/var/log/syslog
#cron.*                  /var/log/cron.log
daemon.*                 -/var/log/daemon.log
kern.*                   -/var/log/kern.log
lpr.*                    -/var/log/lpr.log
mail.*                  @@localhost:5150
mail.*                   -/var/log/mail.log
user.*                   -/var/log/user.log

```

Para que los cambios tengan efecto, reiniciar el servicio con el comando restart.

```
$ sudo service rsyslog restart
```

### 3.8 Instalación Jsvc

La instalación de las librerías que permitan que la aplicación java corra como demonio es muy sencilla, simplemente se debe ejecutar el comando de instalación.

```
$ sudo aptitude install jsvc
```

## 4 MANUAL DE COMPILACIÓN

Para poder realizar la compilación de cada una de las aplicaciones de las que esta compuesta se deben seguir los pasos descritos en los siguientes puntos.

### 4.1 Compilación de aplicaciones para Android

En primer lugar para poder compilar una aplicación Android se tendrá arrancado un dispositivo virtual de Android, para ello en Eclipse se pulsa sobre el icono del ADV Manager.

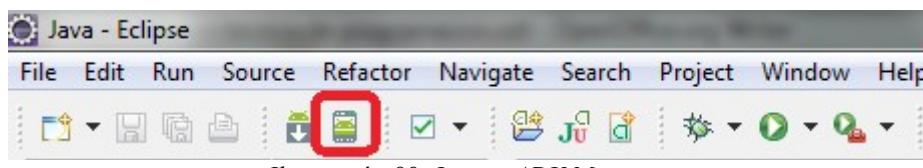


Ilustración 89: Iniciar ADV Manager

Se desplegará el ADV Manager, pulsar New para poder configurar un nuevo dispositivo virtual.

Introducir los datos deseados para la configuración y pulsar Create AVD.

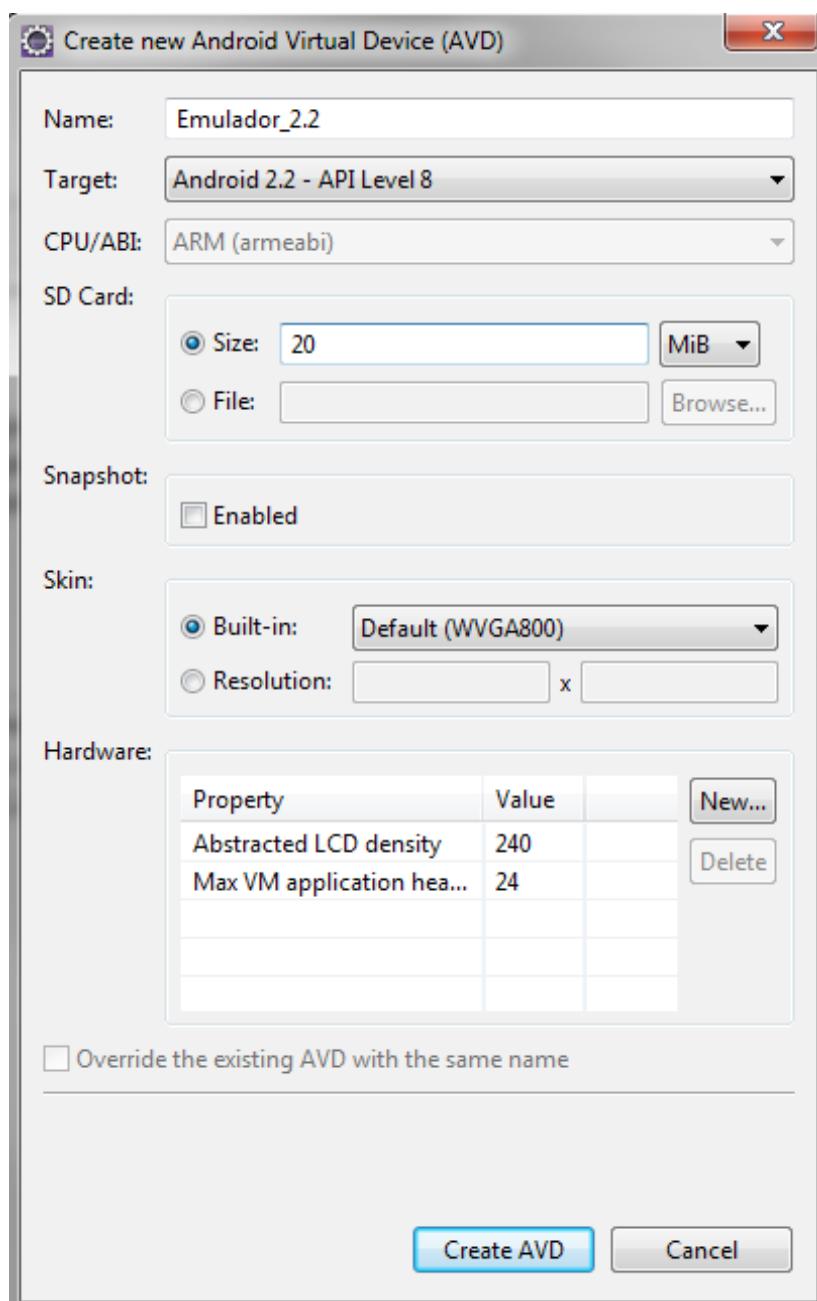


Ilustración 90: Configuración dispositivo virtual Android

Iniciar el nuevo dispositivo creado pulsando el botón Start, una vez iniciado el dispositivo, para compilar la aplicación pulsar el ícono Run, se compilará, instalará y ejecutará en el dispositivo virtual arrancado.

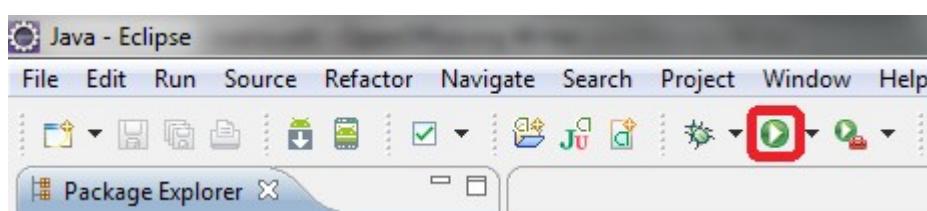


Ilustración 91: Compilación aplicación Android

Una vez compilada y ejecutada la aplicación Eclipse crea en /bin un fichero dex (formato ejecutable de la máquina virtual contenida en Android) y un fichero apk para su distribución (firmado con una clave de depuración de forma automática).

## 4.2 Compilación del Web Service

A la hora de compilar el Web Service, se seguirán los pasos detallados a continuación.

En primer lugar desde el proyecto Notificador en Eclipse, se tiene que crear el contenedor de Web Services, para ello se pincha sobre la carpeta del proyecto con el botón derecho y en el menú desplegable que aparecerá, se pulsa sobre New > Other...

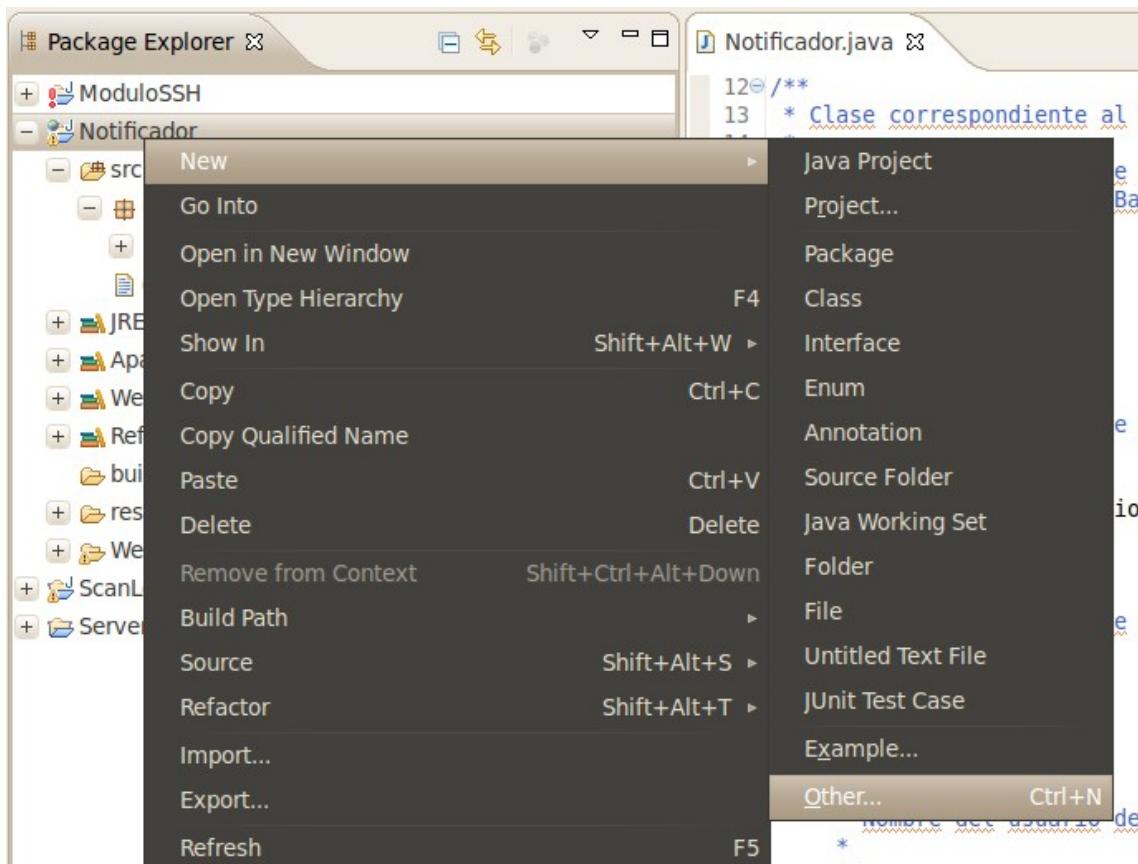


Ilustración 92: Menú desplegable para creación de contenedor de Web Services

Seguidamente buscar la carpeta Web Service y dentro seleccionar Web Service y continuar.

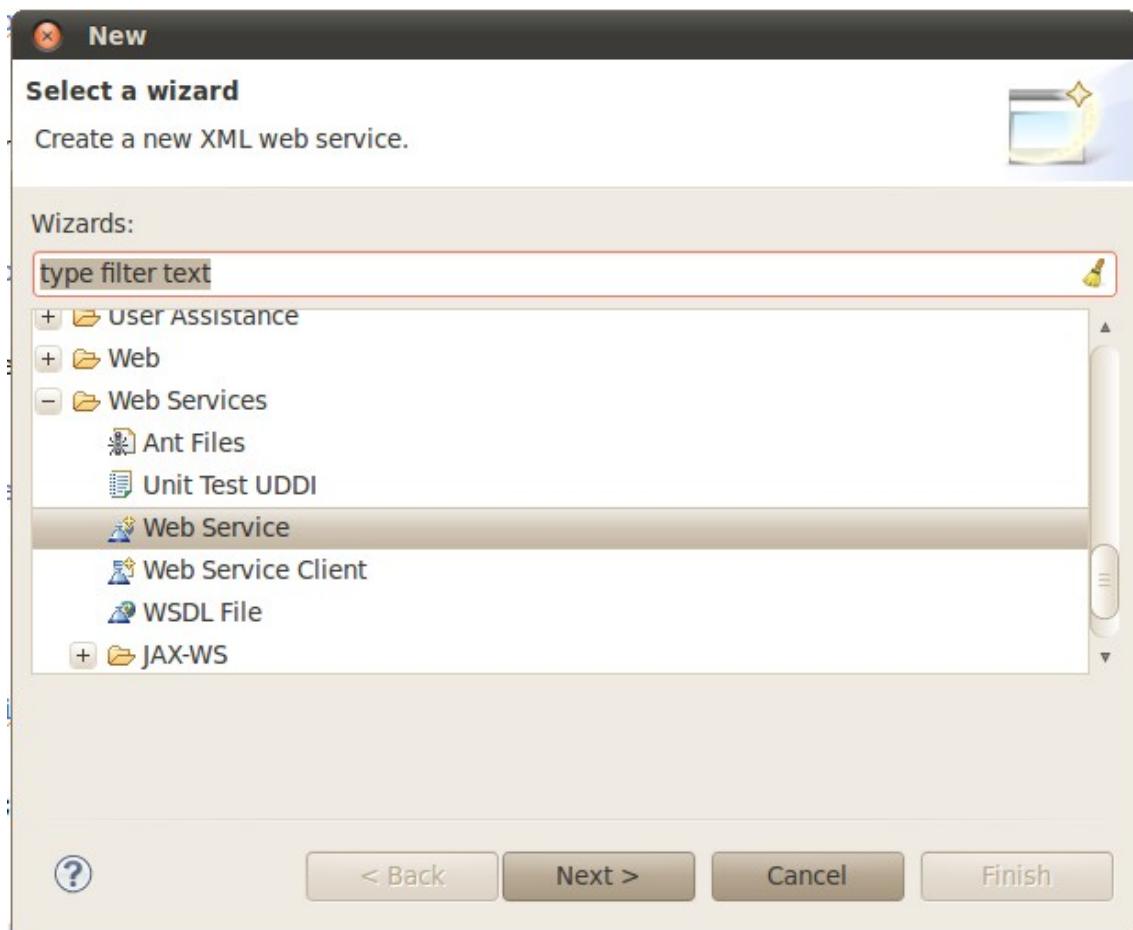


Ilustración 93: Selección de Web Service

En el siguiente paso en el apartado service implementation escoger la clase correspondiente al Web Service, luego se pinchará sobre Web Service Runtime y sobre el listado que se despliega se escogerá Apache Axis2 y pulsar botón OK.

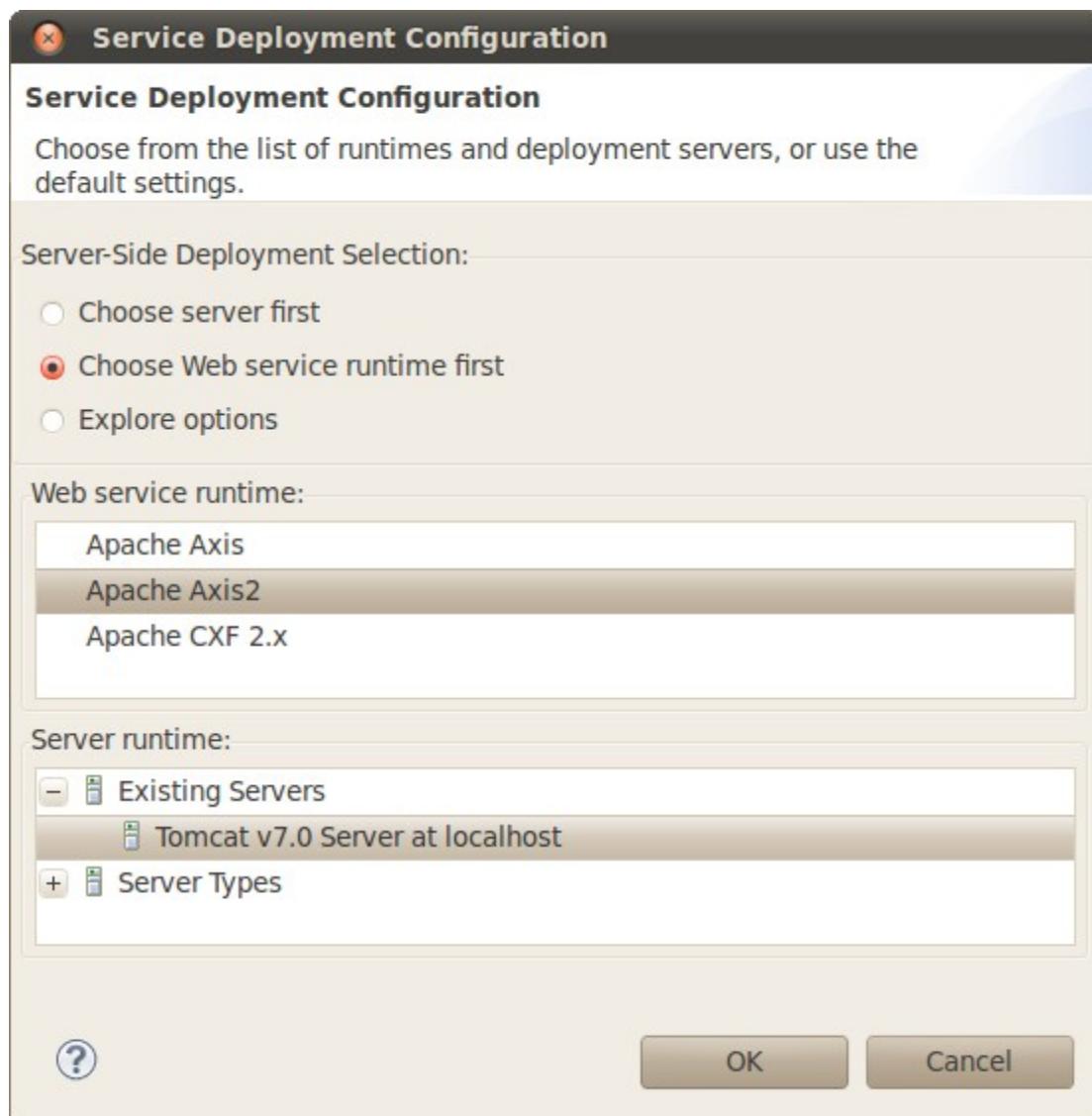


Ilustración 94: Elección Web Service Runtime

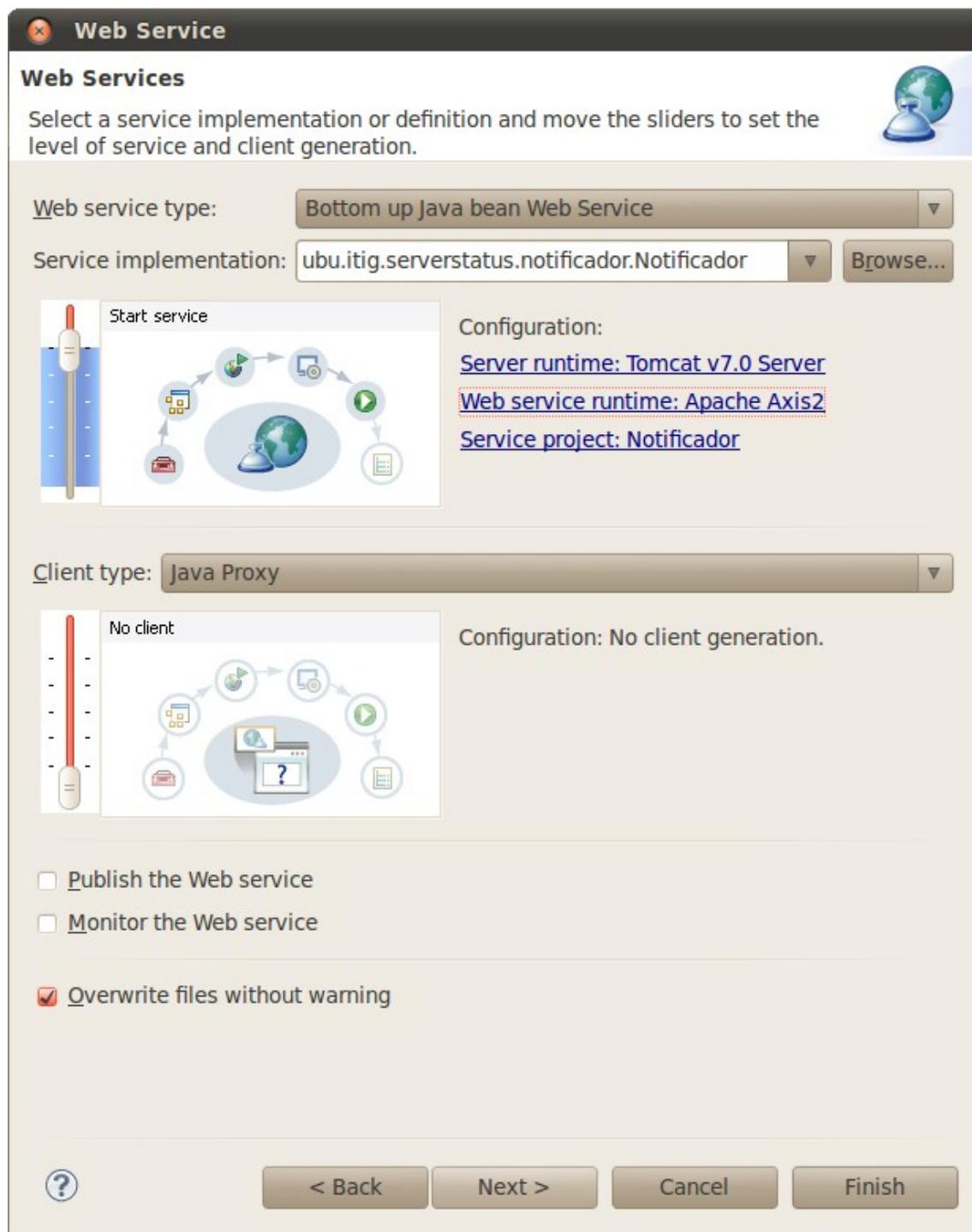


Ilustración 95: Configuración Web Service

Seguido se acepta que se genere un archivo service.xml por defecto, el siguiente paso es arrancar el servidor pulsando el botón Start Server y a continuación una vez iniciado el servidor Finish.

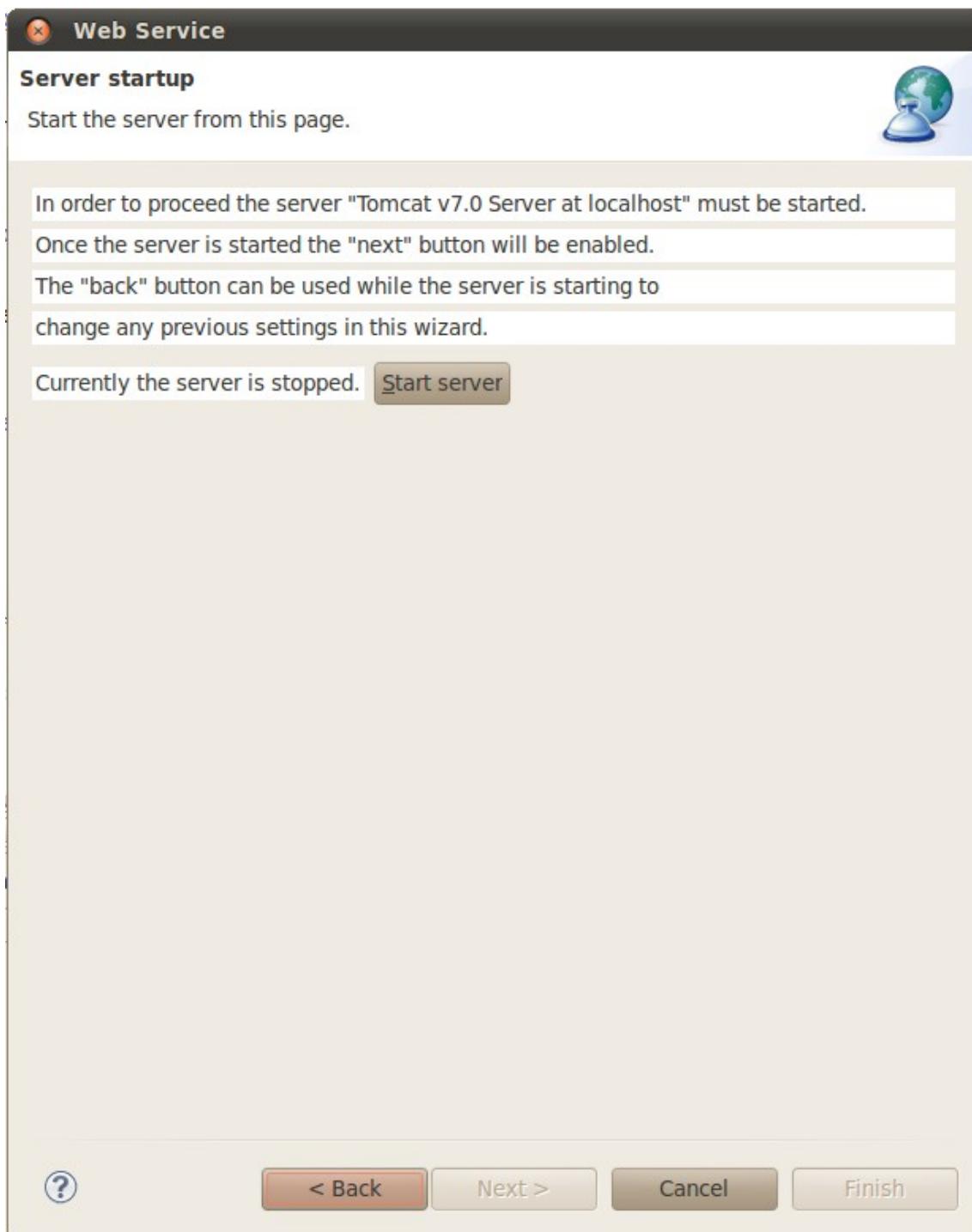


Ilustración 96: Iniciar servidor

A partir de este punto ya se podrá compilar y ejecutar el proyecto, el siguiente paso será, desde el menú desplegable al realizar click con el botón derecho sobre el proyecto pulsar en Run As > Run on server.

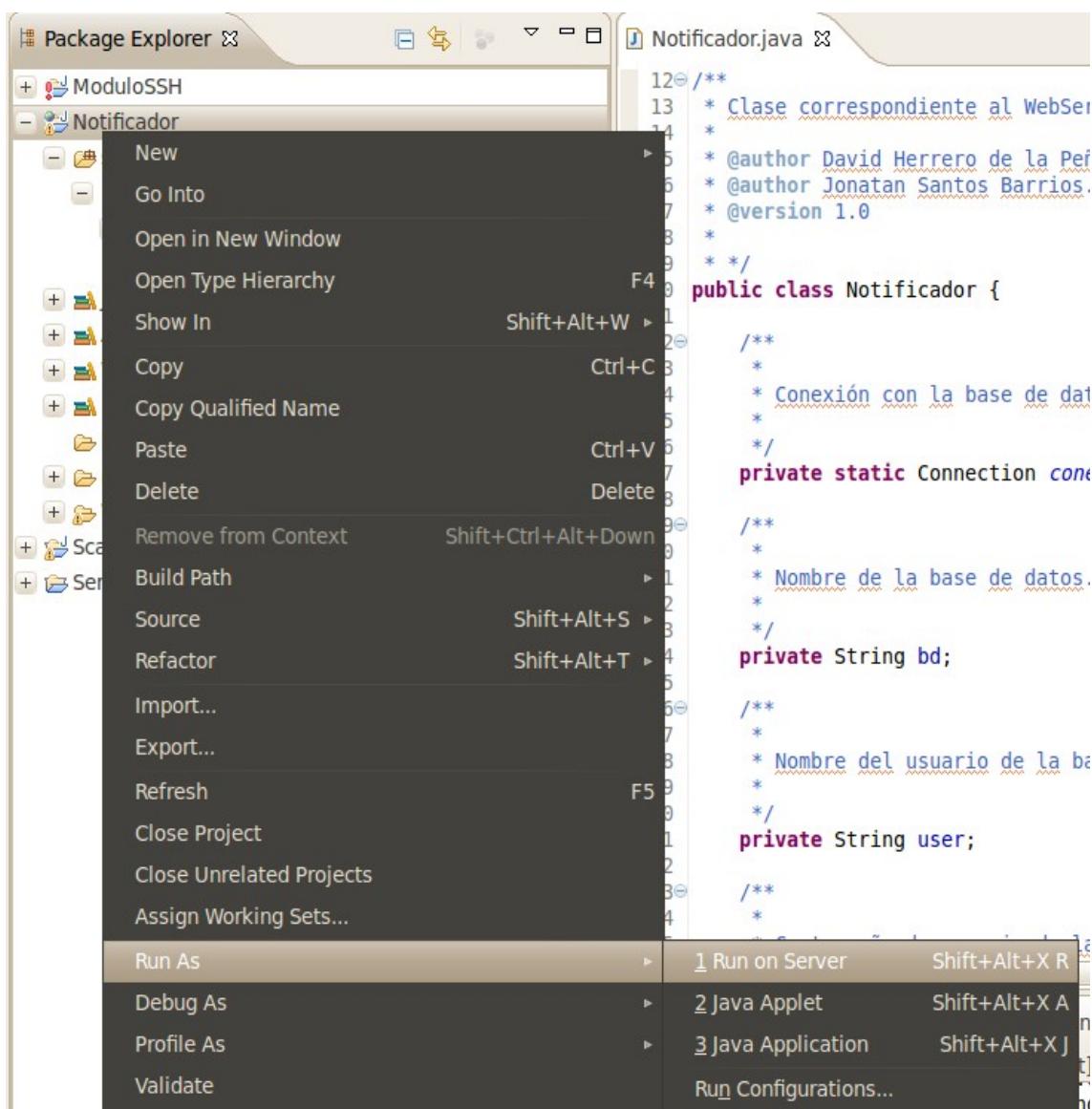


Ilustración 97: Menú desplegable compilación Web Service

A continuación escoger de la lista de servidores el arrancado anteriormente.

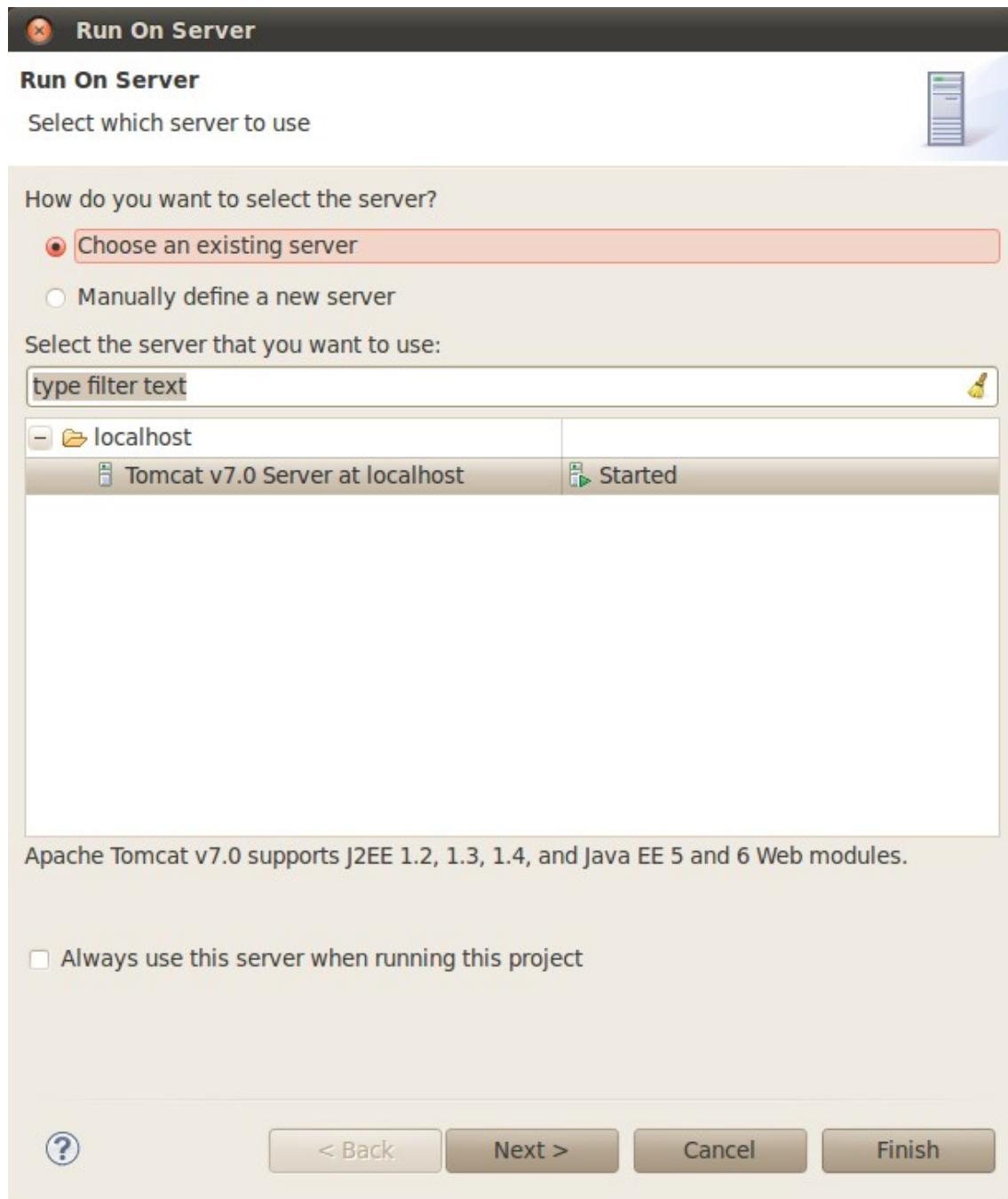


Ilustración 98: Selección de servidor donde correr el Web Service

Y por ultimo añadir la clase que contenga el Web Service y finalizar.

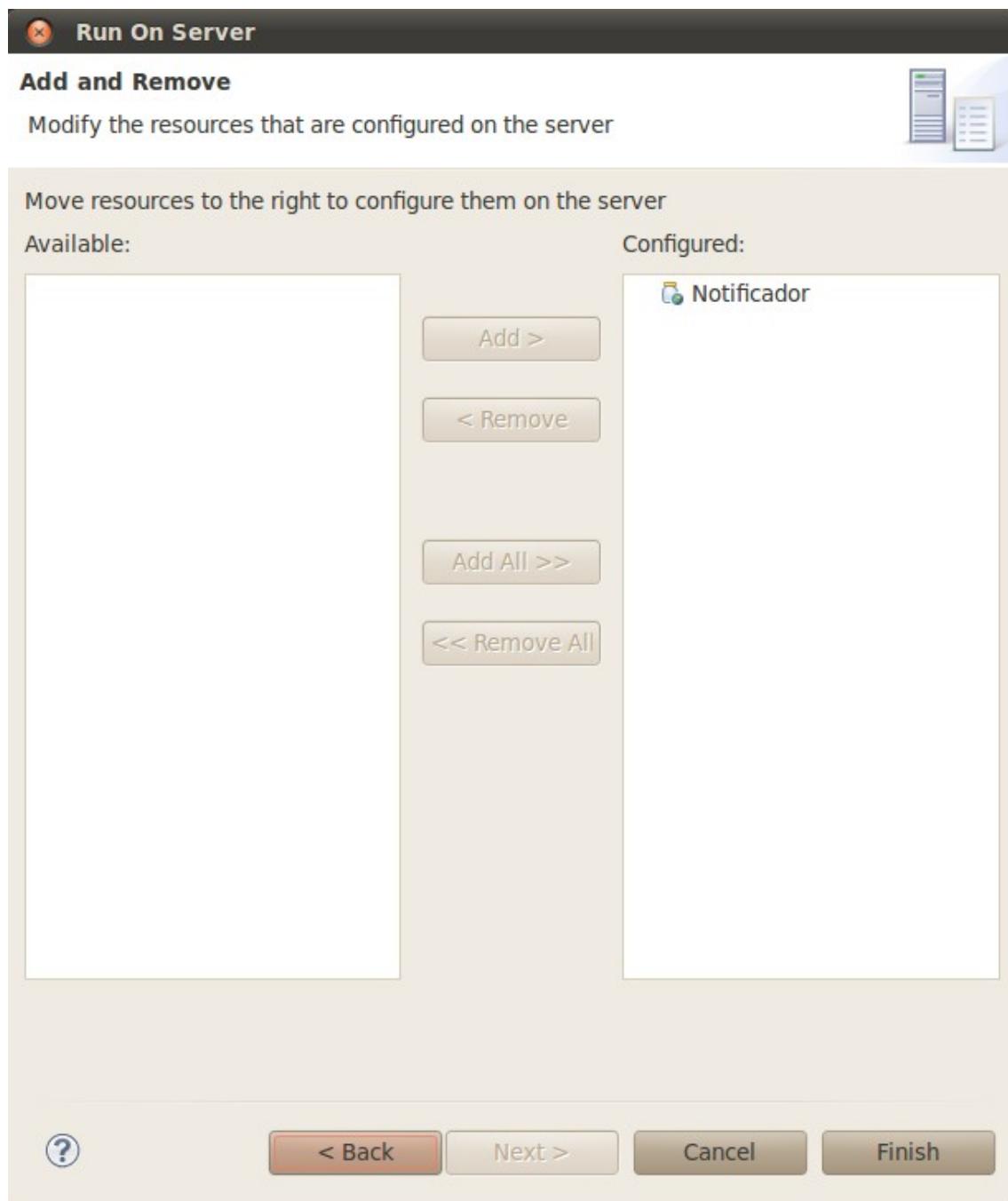
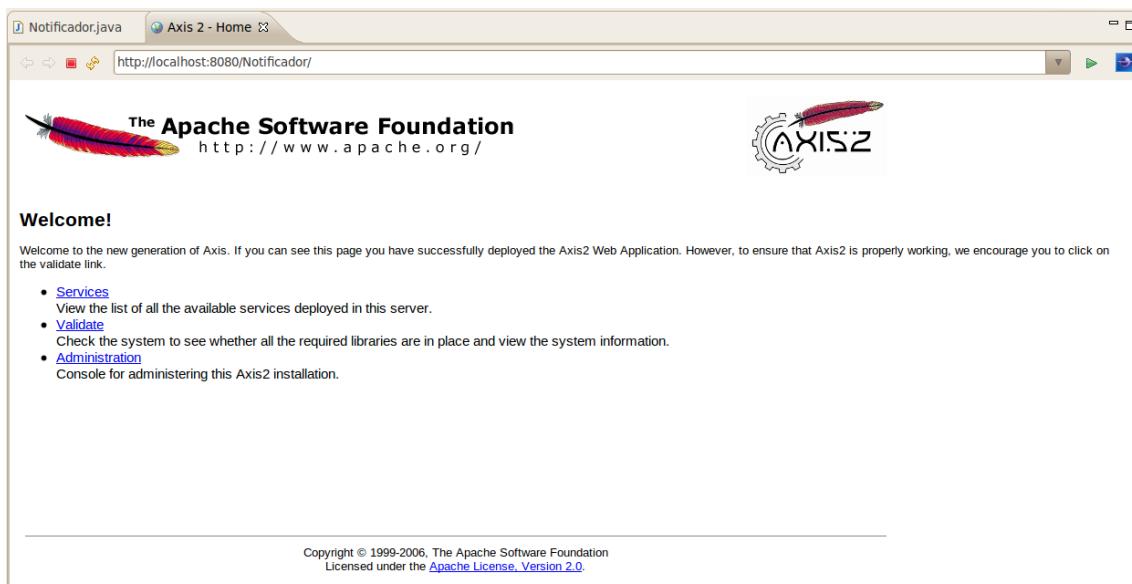


Ilustración 99: Añadir la clase del Web Service

Automáticamente se abrirá una pestaña en Eclipse donde se mostrará la pantalla inicial de Axis 2.



*Ilustración 100: Pantalla principal Apache Axis2*

Para comprobar que el Web Service ha sido publicado correctamente, pinchar sobre Services, aparece un ventana con los Web Service publicados, entre los que aparecerá el añadido y sus métodos públicos.

*Ilustración 101: Listado Web Services publicados*

A la hora de poder crear un paquete para poder instalarlo en cualquier contenedor de servicios web, se exportará un archivo aar, que contendrá el fichero class del Web Service, el conf.properties que contiene información de configuración y el fichero service.xml creado al compilar el proyecto.

Para ello se abrirá el menú desplegable desde el botón derecho y se escoge Export, elegir exportar a un archivo jar y seleccionar los archivos que se encontrarán en la ruta /WebContent/WEB-INF/services/, y elegir la ruta donde guardar el archivo.

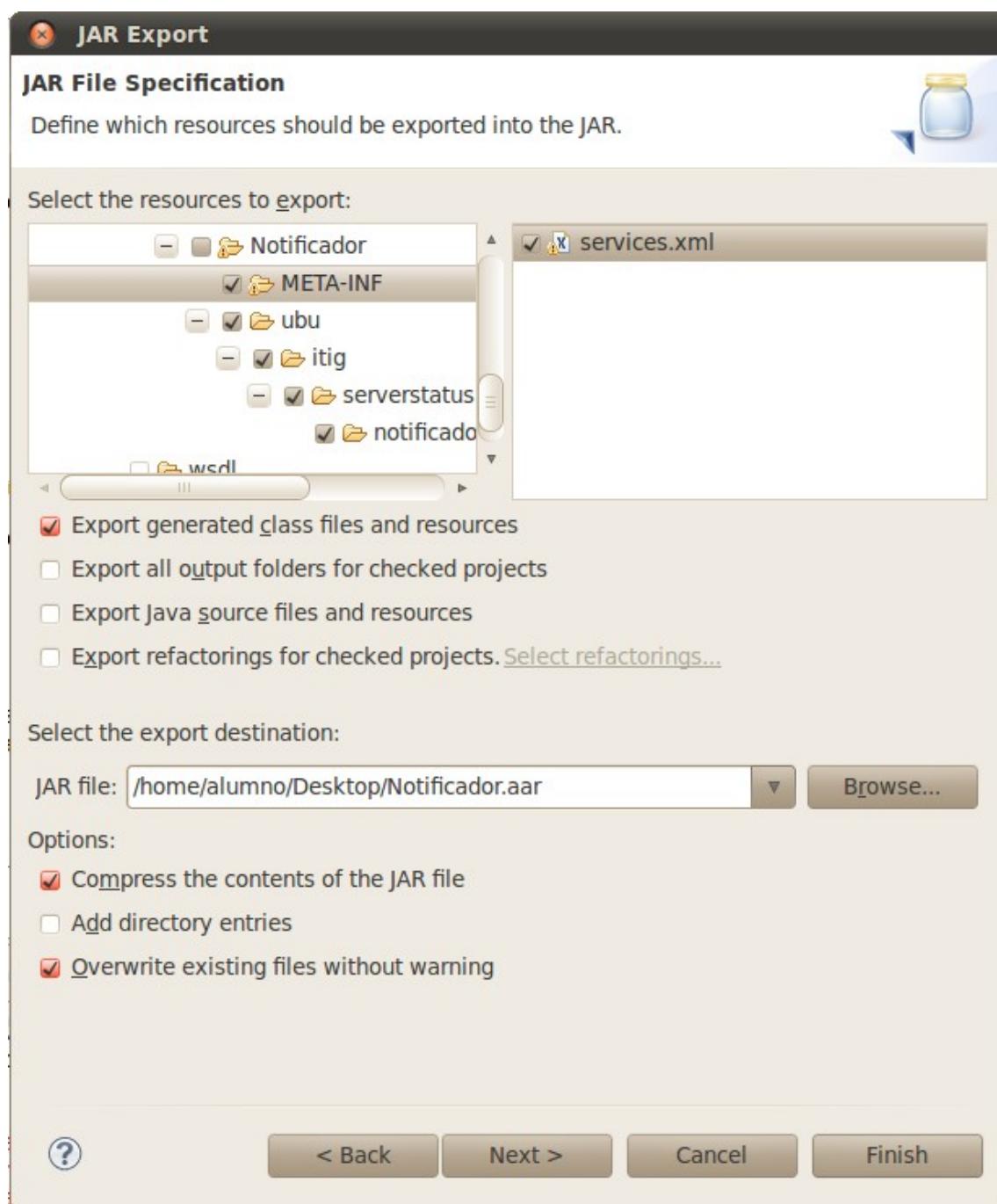


Ilustración 102: Creación archivo aar para Web Service

### 4.3 Compilación del Demonio

Para generar el archivo para poder correr el demonio simplemente tendremos que exportar la clase correspondiente a un archivo jar, se seguirán los mismos pasos que para la exportación en el Web Service, pero en este caso solo se seleccionara el archivo de la clase y el archivo conf.properties.

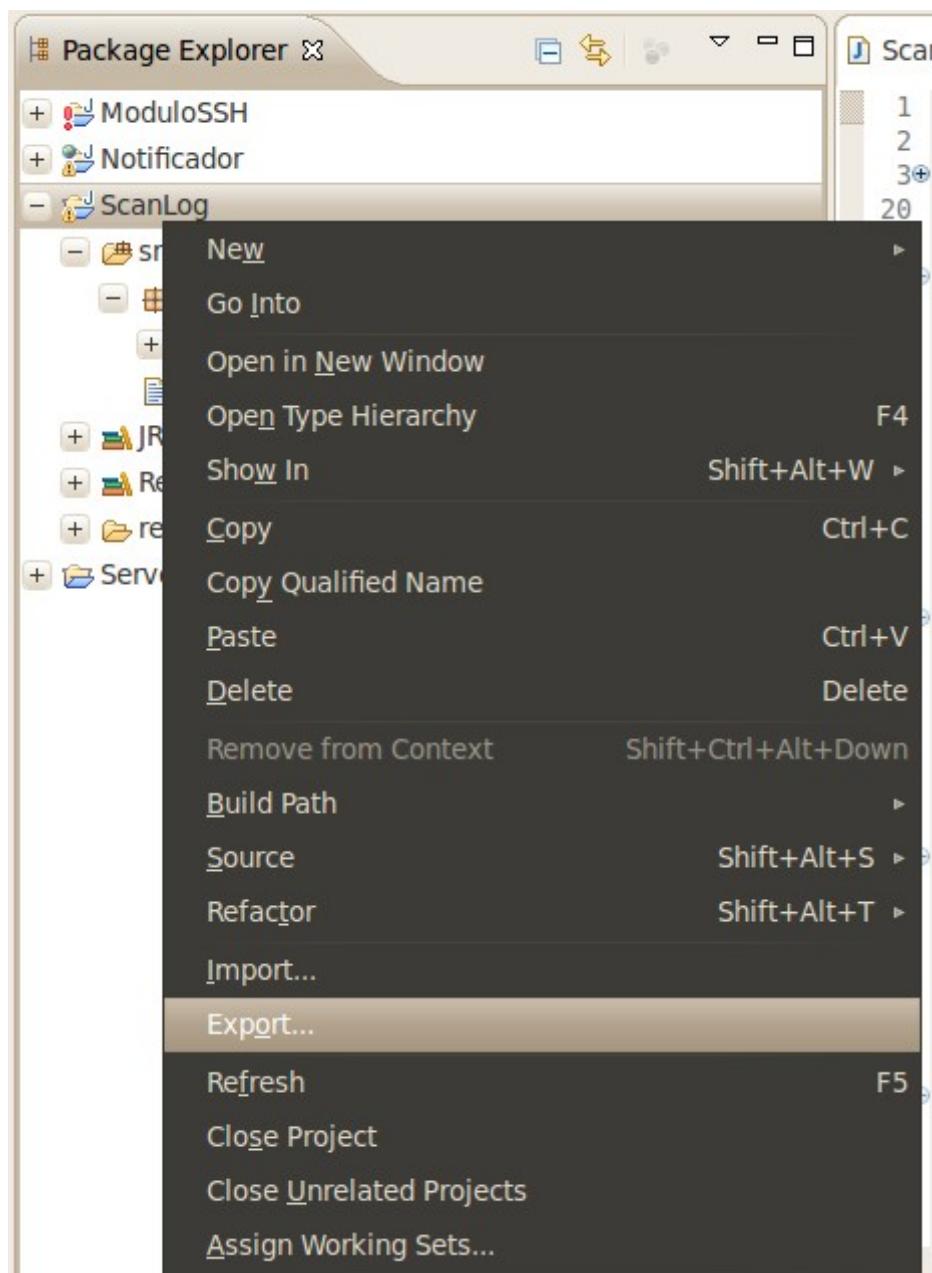


Ilustración 103: Menú desplegable para exportación



Ilustración 104: Selección de exportación a archivo JAR

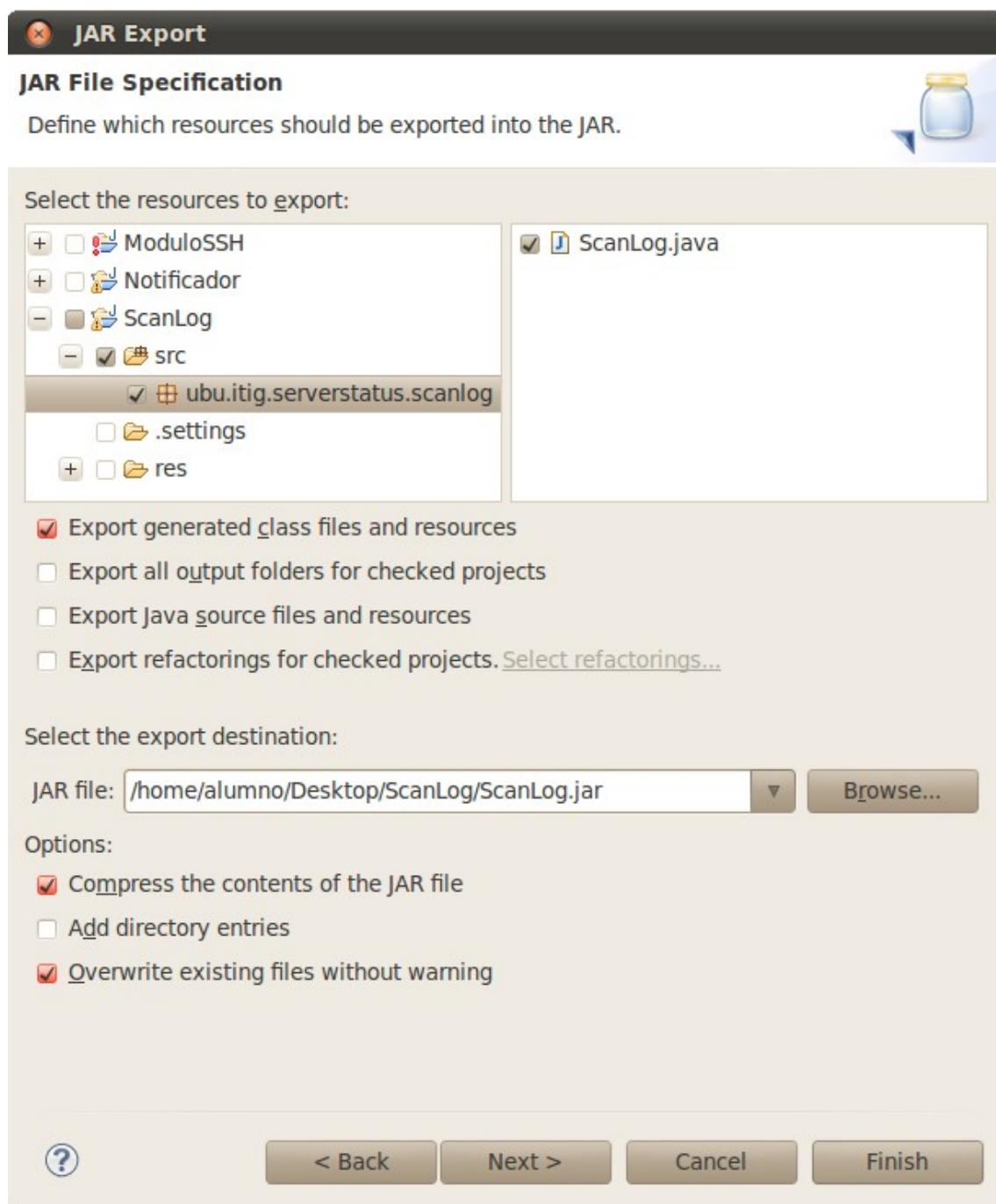


Ilustración 105: Datos de exportación del demonio

## 5 PRUEBAS

### 5.1 Pruebas de integración

Para comprobar la aplicación y la interacción de todas las clases se proponen una serie de pruebas de integración por cada módulo indicando la entrada y salida esperada.

#### 5.1.1 Pruebas sobre el Launcher

Pruebas sobre Launcher			
Prueba	Descripción	Entrada	Salida esperada
P01	Entrar introduciendo la contraseña correcta.	Contraseña correcta.	Ventana principal del Launcher.
P02	Entrar introduciendo una contraseña incorrecta.	Contraseña incorrecta.	Ventana aplicación falsa.

Tabla 48: Pruebas sobre Launcher

#### 5.1.2 Pruebas sobre el Terminal SSH

Pruebas sobre Terminal SSH			
Prueba	Descripción	Entrada	Salida esperada
P01	Conexión con datos de configuración incorrectos.	Datos incorrectos.	Mensaje comprobar datos y conexión.
P02	Conexión sin acceso a la red.	Datos correctos.	Mensaje comprobar datos y conexión.
P03	Conexión correcta.	Datos correctos.	Consola de comandos.
P04	Introducción de comando.	Comando.	Resultado por pantalla.
P05	Añadir favorito.	Datos de conexión.	Añadido servidor al listado de favoritos.
P06	Editar favorito.	Datos modificados.	Modificados datos en el listado de favoritos.
P07	Borrar favorito.	Notificación.	Listado de favoritos sin el favorito eliminado.

Tabla 49: Pruebas sobre Terminal SSH

### 5.1.3 Pruebas sobre el Control de Accesos

Pruebas sobre Control de accesos			
Prueba	Descripción	Entrada	Salida esperada
P01	Conexión con datos de configuración incorrectos.	Datos incorrectos.	Notificación de error en conexión.
P02	Conexión sin acceso a la red.	Datos correctos.	Notificación de error en conexión.
P03	Conexión correcta.	Datos correctos.	Nada, esperando a que se produzcan notificaciones nuevas.
P04	Añadir servidor.	Datos de servidor.	Añadido servidor al listado de servidores.
P05	Editar servidor.	Datos modificados.	Modificados datos en el listado de servidores.
P06	Borrar servidor.	Servidor.	Listado de servidor sin el servidor eliminado.
P07	Descargar notificaciones.	Tipo Notificación.	Listado con las notificaciones.
P08	Aplicar filtro.	Datos del filtro.	Listado con las notificaciones que cumplen el filtro.
P09	Borrar notificación.	Notificación.	Listado de notificaciones sin la borrada.

Tabla 50: Pruebas sobre Control de accesos

### 5.1.4 Pruebas sobre el localizador GPS

Pruebas sobre Localizador GPS			
Prueba	Descripción	Entrada	Salida esperada
P01	No esta activado el gps ni la conexión de datos.	Ninguna.	No se registran las coordenadas.
P02	Esta activado el gps o la conexión de datos.	Ninguna.	Se registran las coordenadas en base de datos.
P03	Configuración correcta de aviso por email.	Datos correctos.	Correo electrónico con las ultimas coordenadas.
P04	Configuración incorrecta de aviso por email.	Datos incorrectos.	Ninguna, se envió el correo a la cuenta errónea.
P05	Configuración correcta de aviso por SMS.	Datos correctos.	SMS con las ultimas coordenadas.
P06	Configuración incorrecta de aviso por SMS.	Datos incorrectos.	Ninguna, se envió el SMS al teléfono erróneo.

Tabla 51: Pruebas sobre Localizador GPS

### 5.2 Pruebas de stress

Las pruebas de stress se realizaran únicamente sobre el Web Service, para ello se hace uso de la aplicación Jmeter [9], una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios.

Se han generado un grupo de hilos con 400 conexiones, para llevarlo a cabo en la aplicación se añadirá sobre el plan de pruebas un grupo de hilos.

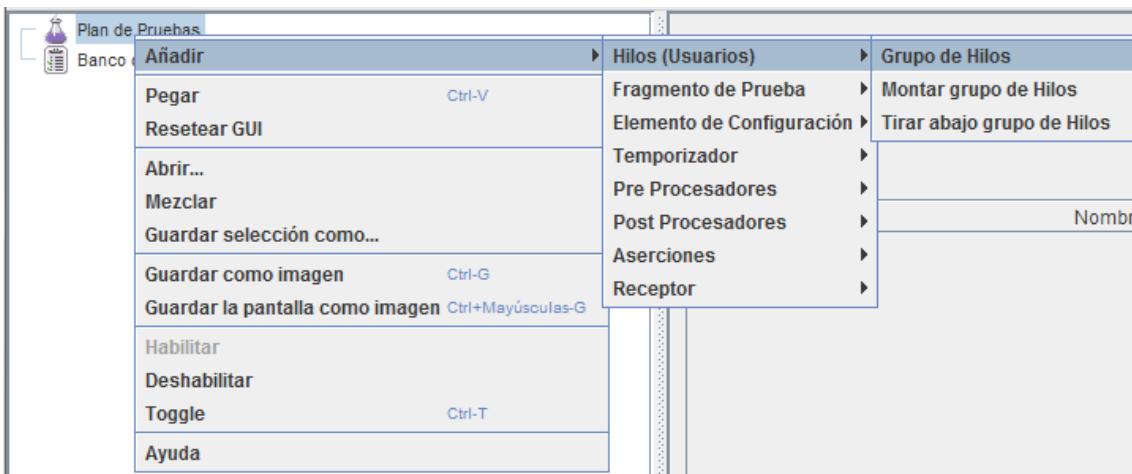


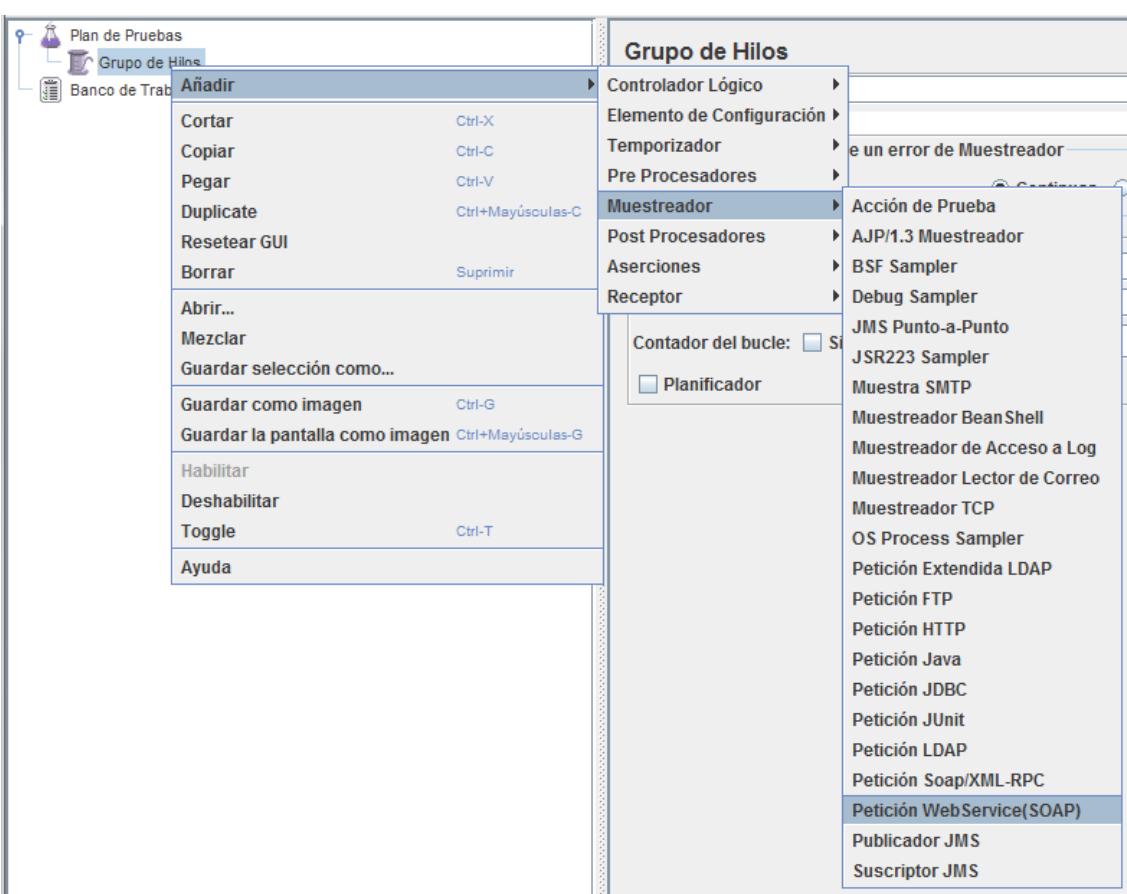
Ilustración 106: Creación de grupo de hilos

Se debe configurar el número de hilos con las conexiones que deseé realizar, en este

caso 400.



El siguiente paso es añadir un muestreador para Web Service (SOAP), esto se realiza desde el grupo de hilos, click con botón derecho y Añadir > Muestreador > Petición Web Service (SOAP).



A continuación se deberá configurar todos los datos del Web Service, en primero lugar se introducirá la dirección WSDL y se pulsará sobre el botón Cargar WSDL, automáticamente se cargarán los métodos disponibles, seleccionar uno, por ejemplo el de hayNotificaciones, y por ultimo introducir el mensaje Web Service.

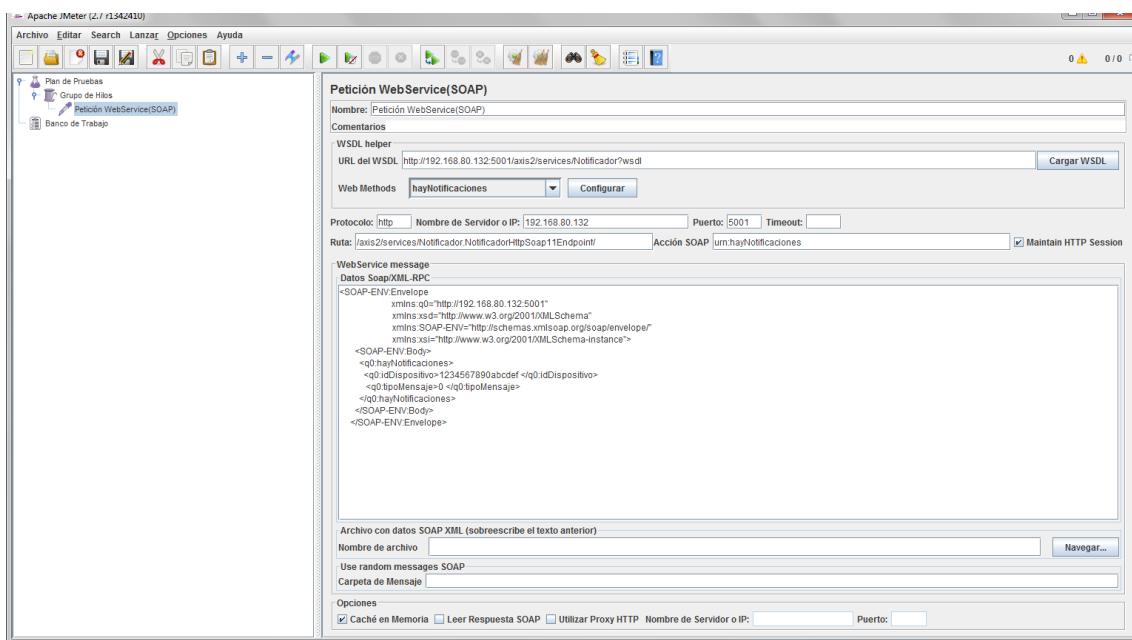


Ilustración 109: Configuración datos Web Service

Para finalizar se añadirá un receptor que recibirá los datos y se podrá realizar el análisis de las pruebas, Añadir > Receptor > Gráfico de Resultados.

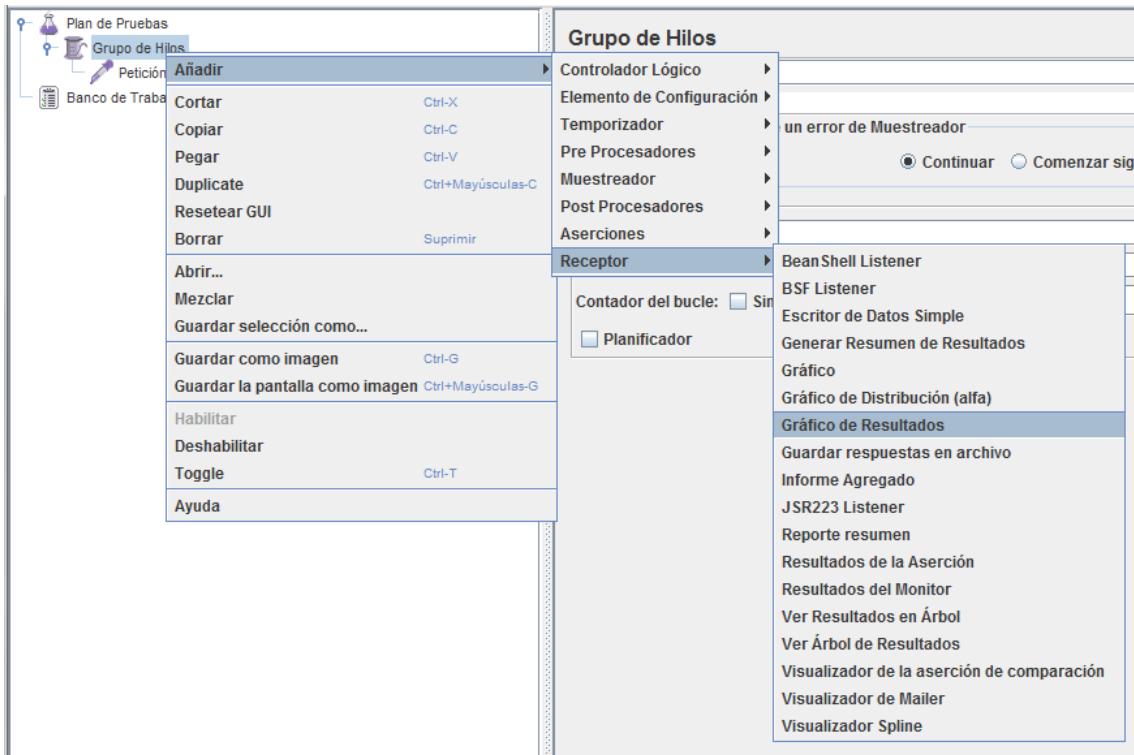


Ilustración 110: Añadir Gráfico de Resultados

Ejecutar las pruebas pulsando el icono de Run de la barra de herramientas, y esperar a que finalicen, automáticamente aparecerá un resumen de los resultados y un gráfico.

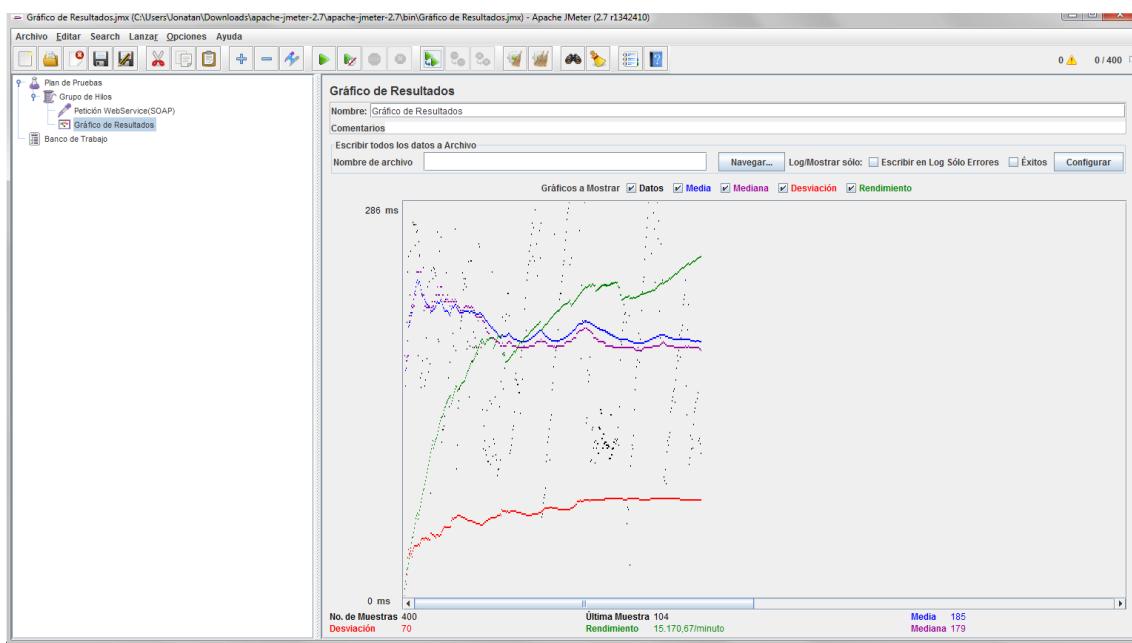


Ilustración 111: Resultado de las pruebas de stress

## Referencias

- [1] Conector MySQL - <http://www.mysql.com/downloads/connector/j/>
- [2] Librerias Jsvc - [http://commons.apache.org/daemon/download\\_daemon.cgi](http://commons.apache.org/daemon/download_daemon.cgi)
- [3] Librerias Jsch - <http://www.jcraft.com/jsch/>
- [4] Java JDK - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [5] Android SDK - <http://developer.android.com/sdk/index.html>
- [6] Eclipse - <http://www.eclipse.org/downloads/>
- [7] Apache Tomcat 7.0 - <http://tomcat.apache.org/download-70.cgi>
- [8] Apache Axis2 - <http://axis.apache.org/axis2/java/core/download.cgi>
- [9] Apache Jmeter - <http://jmeter.apache.org/>



# **UNIVERSIDAD DE BURGOS**

## **ESCUELA POLITÉCNICA SUPERIOR**



**Ingeniería Técnica en Informática de Gestión**

### **Android Server Status**

**Anexo V - Manuales de usuario**

Alumnos      David Herrero de la Peña  
                  Jonatan Santos Barrios

Tutor      Carlos López Nozal  
                  DEPARTAMENTO DE INGENIERÍA CIVIL  
                  Área de Lenguajes y Sistemas Informáticos  
                  Rubén Arribas Barrio

Burgos, 26 de junio de 2012



## **Índice de contenido**

1. Introducción.....	210
2. Manual de instalación.....	210
2.1. Instalación aplicaciones Android.....	210
2.2. Instalación Web Service.....	210
2.3. Instalación Demonio.....	211
3. Manual de usuario.....	212
3.1. Web Service.....	213
3.2. Demonio.....	213
3.3. Cliente Android.....	214
3.3.1. Launcher.....	214
3.3.2. Terminal SSH.....	218
3.3.3. Localizador GPS.....	222
3.3.4. Control de accesos.....	225

## Índice de ilustraciones

Ilustración 112: Web Service publicado.....	211
Ilustración 113: Pantalla Login.....	215
Ilustración 114: Pantalla falsa.....	216
Ilustración 115: Pantalla Principal.....	217
Ilustración 116: ssh manual.....	218
Ilustración 117: ssh favoritos.....	218
Ilustración 118: Consola.....	219
Ilustración 119: Últimos ejecutados.....	220
Ilustración 120: Añadir último.....	221
Ilustración 121: Ejecutar favorito.....	222
Ilustración 122: Preferencias GPS.....	224
Ilustración 123: GPS principal.....	225
Ilustración 124: Control ajustes.....	226
Ilustración 125: Formulario control.....	227
Ilustración 126: Lista servidores.....	228
Ilustración 127: Actualizando notificaciones.....	229
Ilustración 128: Filtro.....	230
Ilustración 129: Borrando notificaciones.....	231

## Índice de tablas

## **LISTA DE CAMBIOS**

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>
1.0	13/06/12	Primera versión: Introducción.
2.0	14/06/12	Segunda versión: Manual de usuario cliente Android.
3.0	16/06/12	Tercera versión: Manual de usuario Web Service y Demonio.

## 1 INTRODUCCIÓN

Este anexo contiene los manuales de instalación con todos los pasos necesarios para hacer funcionar la aplicación, así como los manuales de usuario para familiarizarse con el uso de la aplicación y de las distintas configuraciones y posibilidades de las que dispone, reseñar que el usuario de esta aplicación debe tener conocimientos de administración de servidores para poder hacer uso de ella.

## 2 MANUAL DE INSTALACIÓN

En este manual se indica la forma de instalar todos los componentes necesarios para el correcto funcionamiento de las aplicaciones.

### 2.1 Instalación aplicaciones Android

Para la instalación del cliente Android se necesita un dispositivo que cumpla los siguientes requisitos:

- S.O Android 2.3.3 o superior
- Resolución WVGA de 800x480 o superior
- Chip GPS

Una vez cumplidos todos los requisitos, la forma de instalar consiste simplemente en instalar los **4 apk**, no importa el orden aunque es recomendable instalar el Launcher el último (clientecontrolaccesos.apk, ModuloSSH.apk, LocalizadorGPS.apk, Principal.apk).

### 2.2 Instalación Web Service

Para realizar la instalación del Web Service, es imprescindible que el equipo cumpla los siguientes requisitos software:

- S.O. Ubuntu Server
- Java JRE o JDK
- Apache Tomcat 7.0
- Apache Axis2
- MySQL

A continuación se describen los pasos a seguir para poder realizar la instalación y configuración del Web Service Notificador.

Para instalar en Web Service basta con copiar o mover el archivo aar a la carpeta correspondiente de Axis2.

```
$ sudo mv /home/alumno/Notificador.aar /usr/share/apache-tomcat-7.0.27/webapps/axis2/WEB-INF/services/
```

El Tomcat lo reconocerá sin realizar ninguna operación más, se comprueba que

función correctamente pulsando sobre Services en la página principal de Axis2, y se verá el Web Service y sus métodos.

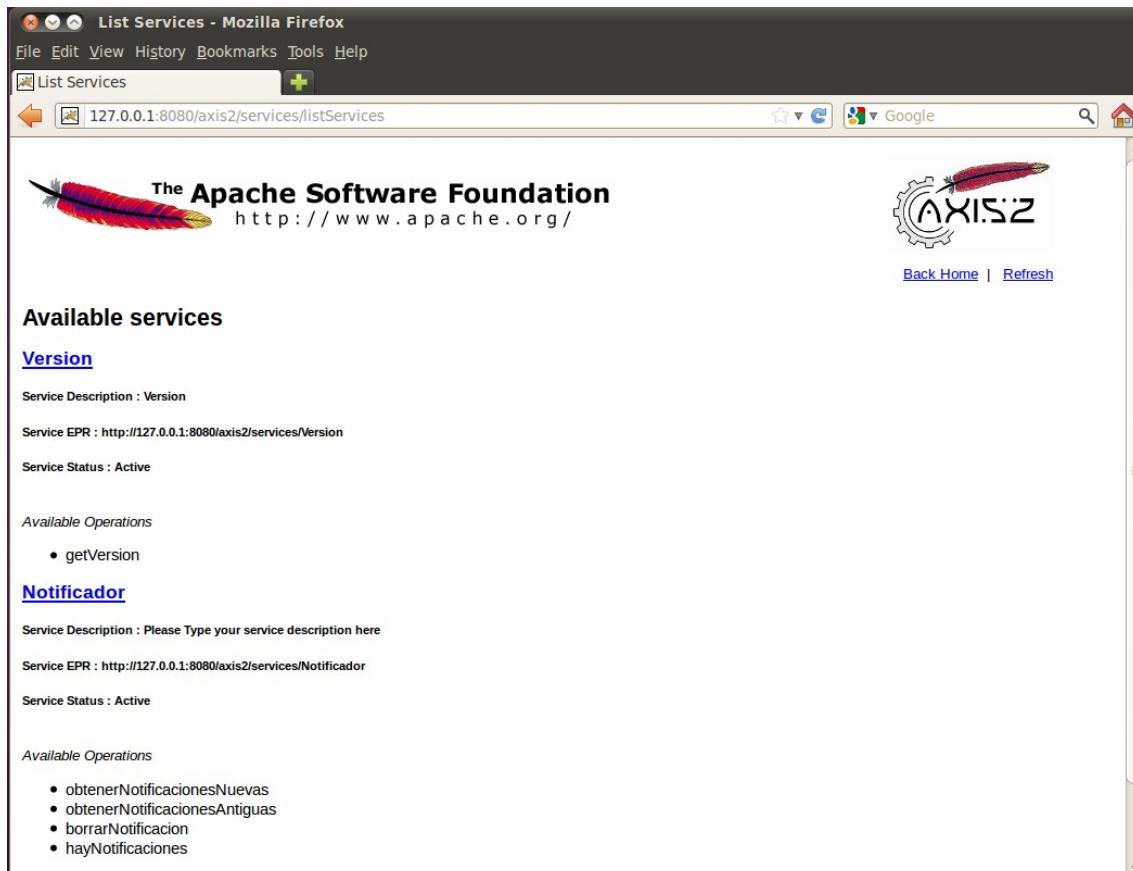


Ilustración 112: Web Service publicado

### 2.3 Instalación Demonio

Para la instalación del demonio, los requisitos software son:

- S.O. Ubuntu Server
- Aplicación de registro de logs (rsyslogd, syslog-ng, sysklogd)
- JSVC

Ahora se describirá el método para instalar y configurar el demonio ScanLog

Para instalar el demonio ScanLog, en primer lugar copiar la carpeta ScanLog a la situación que se deseé.

```
$ sudo mv /home/alumno/ScanLog /usr/share/
```

Luego editar el script para iniciar el demonio,

```
$ sudo gedit /usr/share/ScanLog/ScanLog.sh
#!/bin/sh
```

```
# Setup variables
EXEC=/usr/bin/jsvc
JAVA_HOME=/usr/lib/jvm/jdk1.7.0
CLASS_PATH="/usr/share/java/commons-
daemon.jar":"/usr/share/ScanLog/ScanLog.jar"
CLASS=ubu.itig.serverstatus.scanlog.ScanLog
USER=root
PID=/tmp/scanlog.pid
LOG_OUT=/tmp/scanlog.out
LOG_ERR=/tmp/scanlog.err
```

.....

En este archivo, modificar las líneas de JAVA\_HOME donde se indica la dirección donde está instalado java, CLASS\_PATH según donde se esté instalado cada componente, y PID LOG\_OUT, y LOG\_ERR, según donde se quieran grabar los archivos de log.

Ahora solo hace falta copiar el script que hace que se inicie el demonio, con el inicio del sistema operativo.

```
$ sudo mv /home/alumno/android-server-status /etc/init.d/
```

Abrir el fichero y configurar las direcciones de instalación si es necesario.

```
$ sudo gedit /etc/init.d/android-server-status
```

Para que se pueda arrancar con el inicio del sistema se debe aplicarle permisos de ejecución.

```
$ sudo chmod +x /etc/init.d/android-server-status
```

Y el último paso es crear el enlace correspondiente:

```
$ sudo ln -s /etc/init.d/android-server-status /etc/rc2.d/S99android-server-status
```

Para arrancar el demonio sin reiniciar el servidor ejecutar el siguiente comando.

```
$ sudo /usr/share/ScanLog/ScanLog.sh start
```

### 3 MANUAL DE USUARIO

El proyecto en el que hemos trabajado tiene dos partes bien diferenciadas, el Web Service y el Demonio que se instalan en el servidor Linux, y el cliente que se instala en un terminal Android. Por lo que habrá un manual de usuario para cada una de las partes.

### 3.1 Web Service

Para configurar el Web Service, se cuenta con un archivo conf.properties en que se establecen los datos necesarios para la conexión con la base de datos.

Este archivo se encuentran en el interior del propio archivo del Web Service, es decir en Notificador.aar, a continuación se muestra su estructura.

```
#Servidor de base de datos
dataBaseServer=localhost

#Nombre de la base de datos
dataBaseName=android-server-status

#Usuario de la base de datos
dataBaseUser=alumno

#Contraseña del usuario
dataBaseUser=alumno
```

### 3.2 Demonio

Para configurar el Demonio, al igual que en el Web Service se cuenta con un archivo conf.properties en que se establecen a parte de los datos necesarios para la conexión con la base de datos, otros datos importantes como el puerto de escucha de los mensajes de syslog o los tiempos de actualización, en definitiva datos de configuración.

El archivo de configuración se encuentran en el interior del archivo del Demonio, en ScanLog.jar, la estructura que sigue es idéntica a la del archivo de configuración del Web Service.

```
#Servidor de base de datos
dataBaseServer=localhost

#Nombre de la base de datos
dataBaseName=android-server-status

#Usuario de la base de datos
dataBaseUser=alumno

#Contraseña del usuario
```

dataBaseUser=alumno

#Puerto de escucha de mensajes syslog  
portIn=5150

#Tiempo en milisegundos de comprobación de nuevos mensajes  
tiempoComprobación=1000

#Número de emails enviados por minuto que se considera spam  
numeroMails=6

#Número de intentos de conexión al correo por minuto que se considera ataque  
numeroAtaque=6

### 3.3 Cliente Android

El cliente Android a su vez se compone de 4 aplicaciones, 1 sirve de Launcher y las otras 3 son las que realizan toda la lógica. Para poder acceder a cualquiera de las 3 aplicaciones satélites es necesario el uso del Launcher creado, y si bien no es necesario instalar las 3 para el funcionamiento de la aplicación (son independientes entre ellas) si que es altamente recomendable, para así poder disfrutar de todas las posibilidades que ofrece esta aplicación.

Se dividirá este manual en 4 apartados, uno por cada aplicación, con el objetivo de tener la información mucho más ordenada.

#### 3.3.1 Launcher

Esta aplicación es la más sencilla de todas, no realiza ninguna acción y solo sirve para acceder al resto de aplicaciones.

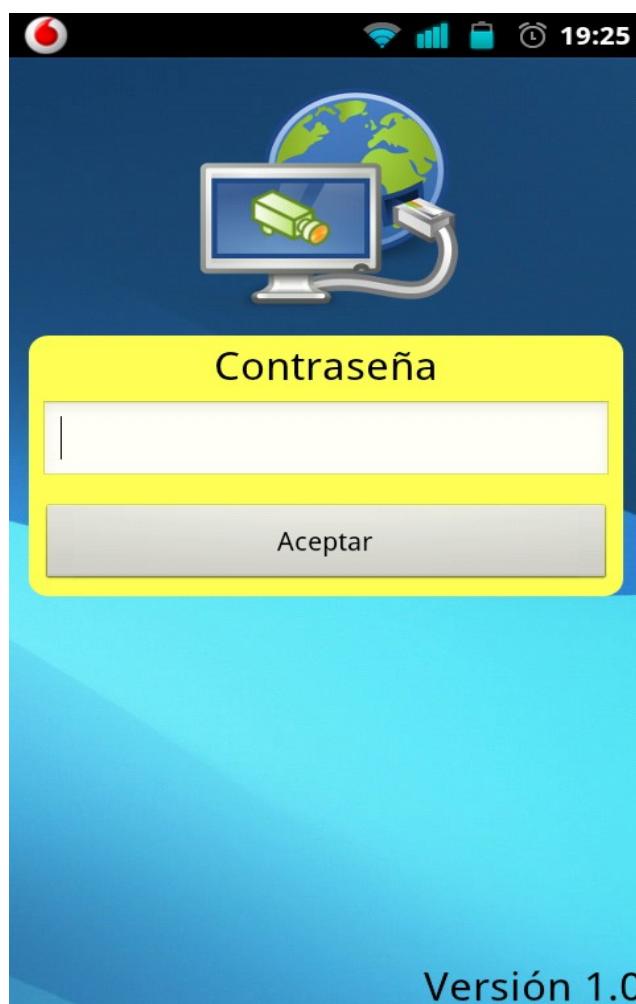


Ilustración 113: Pantalla Login

La primera pantalla que se ve al entrar es una pantalla de login en la que hay que introducir una contraseña.

La contraseña por **defecto** es la **cadena vacía**.

Cuando se introduce la contraseña y se pulsa aceptar pueden ocurrir dos cosas:

- Contraseña correcta: lleva al Launcher real, con los iconos de las otras 3 aplicaciones.
- Contraseña incorrecta: no se informa al usuario de que es una contraseña incorrecta, si no que se lanza una pantalla falta con accesos directos al Android Market y a la página de Google.

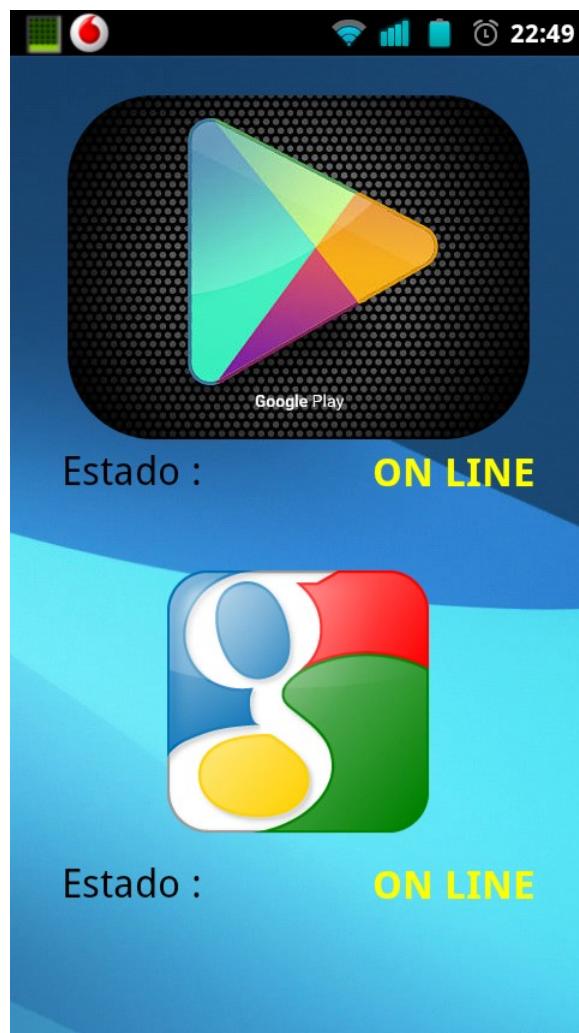


Ilustración 114: Pantalla falsa

Una vez se ha introducido la contraseña correcta y entrado en el verdadero Launcher, se encontrará con 3 iconos, y 3 textos. Además dispone de una pantalla de opciones (desde el Menú > Ajustes) donde podrá cambiar la contraseña.



Ilustración 115: Pantalla Principal

Cada uno de los iconos llevará a la aplicación deseada, la terminal SSH, el localizador GPS o el Control de Accesos.

Los 3 textos que aparecen dan información de los servicios que se están ejecutando en ese momento:

- Localizador GPS: informa de si el servicio está activo, tomando las localizaciones por gps y guardándolas en la base de datos.

- Control SSH: informa de si se está ejecutando el servicio que comprueba si hay notificaciones relacionadas con conexiones ssh en los servidores que se han indicado.
- Control Email: igual que el anterior, pero comprobando notificaciones referentes a envíos de emails e intentos de acceso al email.

### 3.3.2 Terminal SSH

Esta es la parte de la aplicación que permite conectarse mediante ssh a un servidor remoto, y utilizar una terminal emulada para enviar comandos como si se estuviera delante del propio servidor.

Nada más entrar en la aplicación encontrará dos maneras de introducir los datos de un servidor:

- De forma manual: se introducen todos los datos, una vez usados se borrarán de la memoria, para conectarse pulsamos al botón “Conectar”.
- Favoritos: Sirve para añadir los datos de distintos servidores para que se queden almacenados en el teléfono, así la próxima vez que se quiera conectar a esos servidores no hará falta volver a introducir los datos.
  - Si pulsamos en el botón “+” aparecerá el formulario para añadir los datos, si pulsa “aceptar” se guardarán los datos.
  - Para conectarse a un servidor de la lista solo hará falta pulsar sobre él.



Ilustración 116: ssh manual



Ilustración 117: ssh favoritos

En la pestaña de Favoritos hay una opción de menú llamada Clear, si se pulsa se borrarán todos los servidores guardados en el teléfono. Si se realiza una pulsación

**larga** sobre alguno de los servidores de la lista podremos **borrarlo** o **editar** sus datos.

De cualquiera de las formas, cuando intente conectarse a un servidor pueden aparecer problemas :

- No hay conexión o los datos son erróneos
- No se ha conseguido el canal ssh (volver a intentar hasta que se consiga)

Si todo funciona correctamente, conseguirá la conexión ssh y verá la interfaz de la consola.

```
david@ubuntu-server:~$ ls
Descargas hl hola3 [mola      Plantillas prueba5 Vídeos
Documentos hola Imágenes [molaaaaaaaaaaaaaaaa prueba10
prueba6 [will
Escritorio hola2 mola Música      prueba3 Público
david@ubuntu-server:~$ 
david@ubuntu-server:~$ pstree
init─ NetworkManager
      └─3*[VBoxClient]─{VBoxClient}
      ├─VBoxService─7*[{VBoxService}]
      ├─acpid
      ├─atd
      ├─avahi-daemon─avahi-daemon
      ├─bonobo-activat─{bonobo-activat}
      ├─clock-applet
      ├─console-kit-dae─63*[{console-kit-dae}]
      ├─cron
      ├─cupsd
      ├─2*[dbus-daemon]
      ├─2*[dbus-launch]
      ├─dhclient3
      ├─gconfd-2
      ├─gdm-binary─gdm-simple-slav─Xorg
      ├─gdm-session-wor─gnome-
      └─session─bluetoo+
      └─{gdm-simple-slav}
      └─{gdm-session-wor}
      └─{gdm-simple-slav}
```

Ilustración 118: Consola

Para introducir comandos se escribirán en la caja de texto y pulsando el botón play o el enter del teclado. La respuesta del servidor aparecerá en la pantalla.

Todos los comandos que se usan en la sesión se almacenan para usos posteriores.

Si pulsa sobre el menú aparecen varias acciones, en orden son :

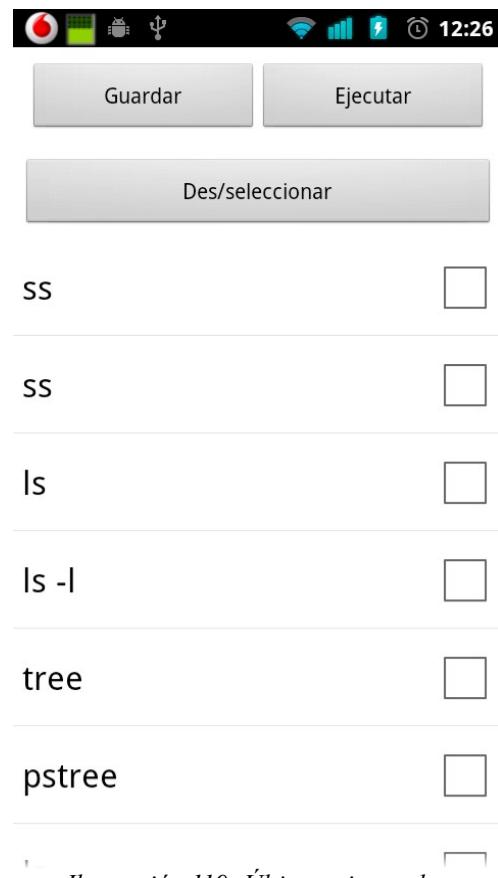


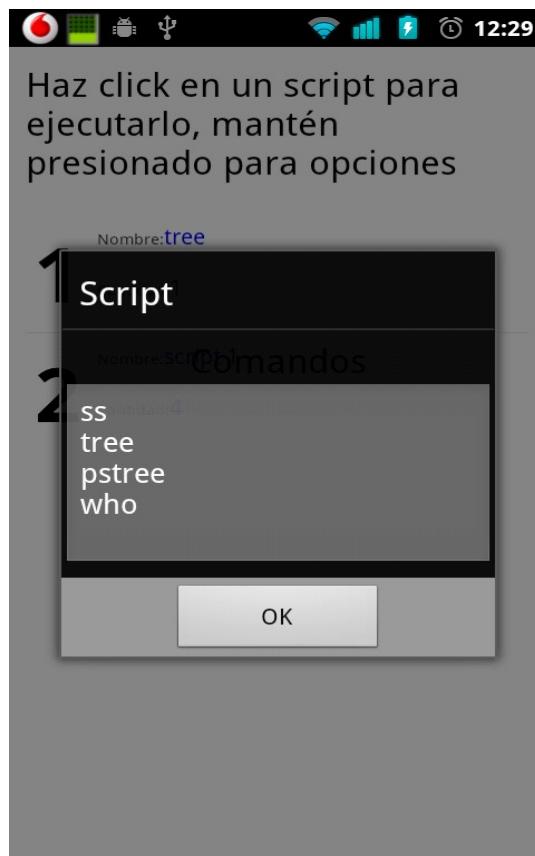
Ilustración 119: Últimos ejecutados

- Últimos ejecutados: muestra una pantalla con los comandos ejecutados en la sesión actual. Puede seleccionar los que quiera para guardarlos en la base de datos o ejecutarlos ahora mismo. En cualquier caso se guardarán en orden descendente y se ejecutarán uno detrás de otro.
- Reconectar: Si se pierde la conexión, o el servidor no responde a los comandos puede ser porque se ha perdido el canal de comunicación, al pulsar esta opción se intentará volver a crear.



Ilustración 120: Añadir último

- Añadir último: Guarda el último comando en la base de datos con el nombre que queramos.
- Borrar: Elimina el texto de la consola.



*Ilustración 121: Ejecutar favorito*

- Ejecutar favorito: Se accede a todos los comandos guardados mediante “añadir último” y “últimos ejecutados” en la base de datos. Para ejecutar el script pulsamos sobre él, aparecerá una pantalla indicando todos los comandos que contiene y si pulsamos “OK” se ejecutará.
- Enviar Ctrl-C: envíá esa orden, útil para detener procesos como “top”
- Cerrar: cierra la sesión y vuelve a la pantalla anterior

Es muy importante indicar que para cerrar la sesión **no** se deben usarse comandos **logout**, **exit** o similares, si no que se debe utilizar la opción de **Cerrar** o pulsar la **tecla atrás** de nuestro teléfono.

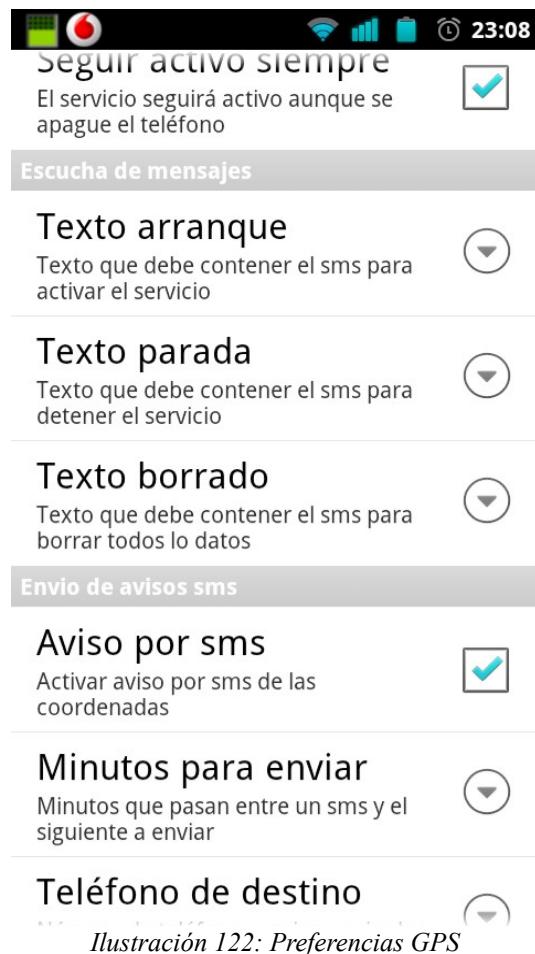
### 3.3.3 Localizador GPS

La aplicación GPS tiene una interfaz muy sencilla, ya que la mayor parte del trabajo la realizan servicios que son invisibles para el usuario.

La parte más importante de esta aplicación es la pantalla de ajustes (Menú > Ajustes). Las opciones a configurar son :

- Seguir activo siempre: si se marca, una vez iniciado el servicio para buscar localizaciones (mediante el sms o con el botón de la pantalla principal) este se volverá a arrancar automáticamente aunque el teléfono se apague. Es recomendado marcar esta opción.

- Texto arranque: es el texto que debe contener el sms para iniciar el servicio de localización gps.
- Texto parada: es el texto que debe contener el sms para detener el servicio de localización gps.
- Texto borrado: es el texto que debe contener el sms para borrar los datos de las otras dos aplicaciones (terminal ssh y control de accesos).
- Aviso por SMS: si se activa se enviarán sms con la última posición de GPS obtenida y la fecha.
- Minutos para enviar: minutos tras los que se enviará un nuevo sms con nuevos datos.
- Teléfono de destino: número de teléfono al que se le enviará el sms.
- Aviso por email :si se activa se enviará un email con TODAS las localizaciones gps que se han ido guardando, junto con la fecha a la que se obtuvieron.
- Minutos para enviar: análogo al sms.
- Correo del destinatario: correo electrónico al que se enviarán los datos.
- Correo de origen: dirección de correo electrónico que se usará para enviar el email, es necesario que sea una cuenta de **google**. Ej: [origen@gmail.com](mailto:origen@gmail.com).
- Contraseña: contraseña del correo de origen.



Se recomienda configurar todos los campos la primera vez que se instale la aplicación.

Las otras tres opciones del menú son :

- Borrar BD: borra todas las localizaciones de la base de datos.
- Enviar sms: envíá un sms con la última posición conocida al número de teléfono que se haya indicado en las preferencias.
- Enviar email: se envíá un email a la dirección que se haya indicado en las preferencias. Se puede configurar la cantidad de localizaciones que se desean enviar.

En la pantalla aparecen 3 botones, cada uno con un ícono distinto, las funciones que realizan cada uno son :

- Botón de mapa: muestra en un mapa la posición gps que se muestra en los campos de texto. Si en los campos de texto pone “desconocido” es que no se ha cargado ninguna localización y por lo tanto no se mostrará nada.
- Botón actualizar: sirve para cargar la última posición de gps almacenada en la base de datos y mostrarla en los campos de textos. Es un paso necesario para

poder visualizarla en el mapa.



*Ilustración 123: GPS principal*

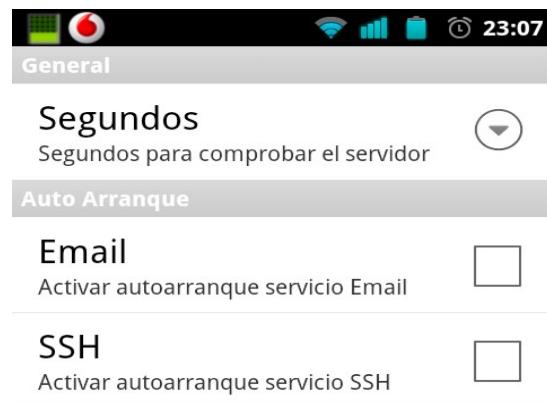
- Botón encendido/apagado: realiza la misma acción que si recibe un sms con el texto para arrancar. El botón arranca el servicio que se ocupa de obtener las localizaciones, y además arranca los servicios de envío de sms y de email si así está indicado en las preferencias. Tener cuidado ya que si se pulsa el botón y está activada la opción de envío de sms se enviarán mensajes con el consiguiente consumo de saldo.

### 3.3.4 Control de accesos

Esta es la aplicación que se encarga de monitorizar todos los servidores a los que nos queremos “suscribir”. La aplicación consta de 3 pestañas principales :

- SSH: aquí están todos los servidores a los que se puede monitorizar los accesos por ssh.
- Email: idéntico al anterior pero monitorizando el correo.
- Notificaciones: aquí se muestran todas las notificaciones que nos han podido enviar los servidores.

En la pantalla de ajustes (Menú > Ajustes) podrá configurar los segundos que pasas entre una comprobación y otra, además de si quiere que la monitorización ssh y email arranquen cuando se encienda el teléfono.



*Ilustración 124: Control ajustes*

Tanto en la pestaña ssh como email se muestra una lista de servidores, para añadir uno solo tendrá que clickar el botón “+” y llenar el formulario.



Ilustración 125: Formulario control

La opción puerto indica el puerto por el que está escuchando el Web Service, y que se habrá indicado en la instalación de Apache Tomcat del servidor.

La opción “iniciar al encender” indica si se empezará a monitorizar ese servidor cuando en ajustes se haya configurado el arranque automático, es decir, en ajustes se puede configurar el arranque automático de ssh, el servicio se iniciará pero no hará llamadas a ningún servidor que no esté activado (botón on/off que aparece en cada servidor), en cambio si ponemos el “arranque automático” ese servidor se activará solo por lo que el se empezarán a realizar consultas al mismo nada más encender el teléfono.

Imagine que ya ha añadido un servidor a la lista, para empezar a monitorizar debe apretar el botón que pone “NO”, cambiará de color y se mostrará un mensaje indicando que la alarma se ha encendido. Si no ha tocado nada se dará cuenta de que no se recibe ninguna notificación, esto es debido a que, aunque el servicio esté iniciado, todavía no ha indicado a qué servidores es necesario consultar.

Cada servidor de la lista tiene a su vez un botón de on/off, si está a off, no se realizarán consultas a ese servidor, pero si le pone a on si.

#### IMPORTANTE

Por lo tanto, para empezar a monitorizar un servidor son necesarias dos cosas:

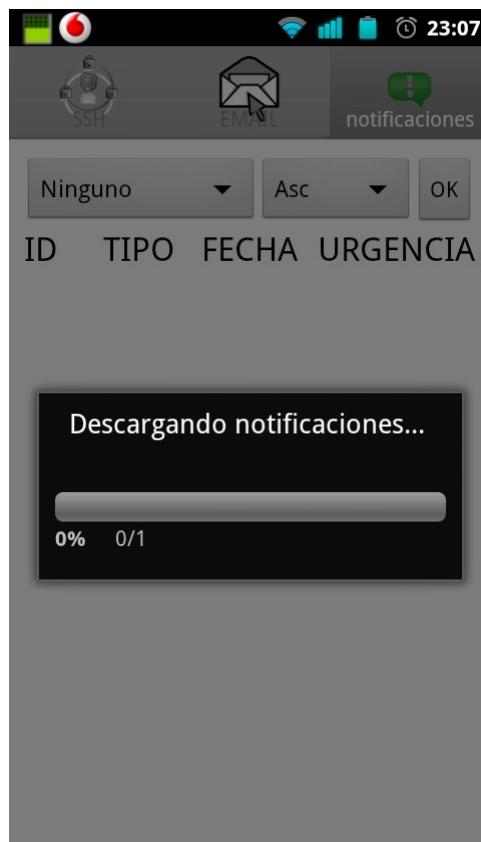
- Servicio arrancado
- El botón on/off del servidor de la lista debe estar activado.



Ilustración 126: Lista servidores

El mecanismo es el mismo en las dos pestañas, de sms y de email.

Cuando alguno de los servidores a los que este suscrito envíe una notificación esta aparece en la barra de notificaciones del teléfono, si pulsa sobre ella le llevará a la pestaña de notificaciones, donde automáticamente se actualizará para descargar las últimas notificaciones de todos los servidores a los que está suscrito (tienen su botón a on) aunque los servicios de monitorización no estén arrancados. Es decir, puede pulsar sobre una notificación de ssh, pero se descargarán todas, ssh y email, aunque el servicio de control de email esté apagado, pero solo se descargarán de aquellos servidores en los que el botón esté a on.



*Ilustración 127: Actualizando notificaciones*

Una vez dentro de la pestaña de notificaciones podrá realizar varias acciones con las notificaciones. Mediante los botones de arriba podrá ordenarlas según el criterio que prefiera.

Si pulsa sobre una notificación se mostrará el texto que se guardó en el log del servidor, con el mensaje que originó la notificación.

Si realiza una pulsación larga sobre una notificación podrá borrarla, no se borrará solo del teléfono, si no que también se borrará de la base de datos del servidor, por lo que si no tiene conexión con el servidor no se podrá realizar la acción.

Esta pantalla tiene varias opciones desde el botón menú :

- Filtro: se abre una nueva pantalla para fijar los criterios de filtrado, una vez que se eligen los criterios y se pulsa el botón OK vuelve a la pestaña de Notificaciones pero solo se visualizarán las notificaciones que cumplan los criterios del filtro.

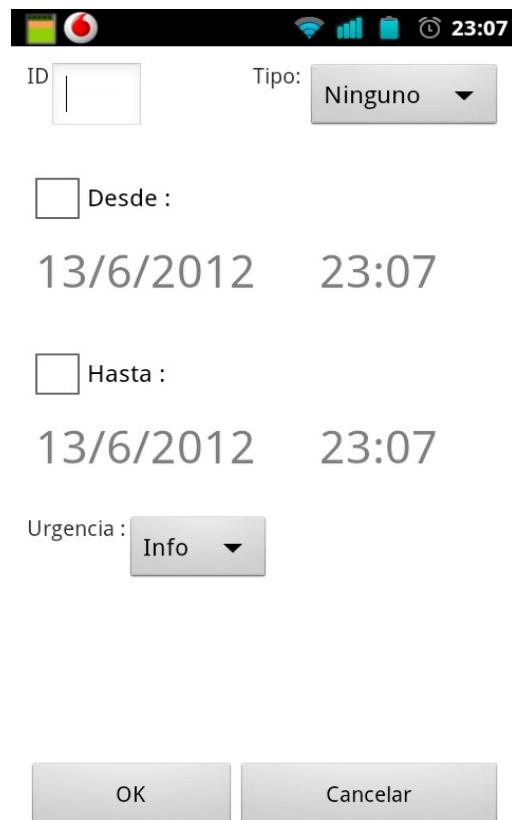
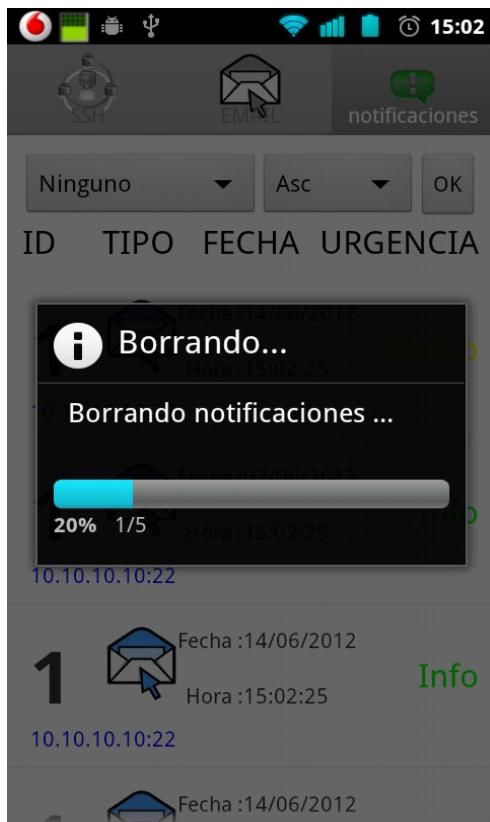


Ilustración 128: Filtro

- Quitar filtro: sirve para quitar un filtro y volver al anterior, se pueden poner filtros unos encima de los otros, y mediante esta opción se vuelve al estado anterior.
- Ajustes: lleva a la pantalla de preferencias.
- Actualizar: realiza la misma acción que cuando se hace click sobre una notificación de la barra de notificaciones del teléfono. Se descarga las nuevas notificaciones que puedan existir y las **añade** a las que se están visualizando ahora. Por razones de consistencia no se puede actualizar cuando hay filtros aplicados.
- Descargar todo: descarga todas las notificaciones todos los servidores activados que han sido guardadas previamente. Cuando se descargan nuevas Notificaciones pasan a ser Notificaciones guardadas (las que se descargan con esta opción) y solo se borran si así se lo indicamos.

- Borrar todo: Borra todas las notificaciones que estén ahora mismo en pantalla, no solo las borra del teléfono, si no que también las borra de los servidores remotos, por lo que es necesario tener acceso a ellos. Esta opción es interesante de usar en conjunto con los filtros, por ejemplo, se usa un filtro para visualizar solo notificaciones de tipo ssh, luego pulsar la opción de “borrar todo” y se borrarán todas las notificaciones de tipo ssh de todos los servidores.





## **AGRADECIMIENTOS**

*En primer lugar agradecemos a Carlos López Nozal y a Rubén Arribas Barrio la oportunidad que nos han brindado para poder llevar a cabo este proyecto y aprender de él y, sobre todo, por la confianza depositada en nosotros y su inestimable interés personal en que conocieramos nuevas herramientas y adquiriesemos nuevos conocimientos, facilitandonos información sobre diversas posibilidades para ponerlos en práctica.*

*A todos los profesores, ya que sin sus clases, apuntes y tiempo invertido en nosotros a lo largo de estos años, no hubiesemos podido llevar a buen puerto el proyecto.*

*Por último, a todos nuestro familiares, amigos y compañeros, porque sin todos sus consejos y apoyo habría sido mucho más difícil la realización de este proyecto.*

*Muchas gracias a todos.*