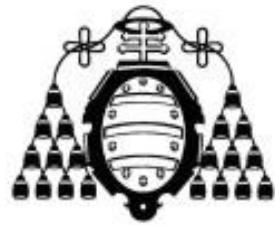


UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

“OCAM – OUTDOOR COLECTIVE ACTIVITIES MONITORIZATION”

DIRECTOR: Alberto Manuel Fernández Álvarez

AUTOR: Víctor Castaño Gutiérrez

**Vº Bº del Director del
Proyecto**

Agradecimientos

Me gustaría mencionar aquí a todas las personas que han intervenido en alguna manera a que este proyecto se haya realizado.

En primer lugar, a Alberto, tutor del proyecto, por el tiempo que ha dedicado, la ayuda, ideas y paciencia durante todos estos meses.

A todos mis compañeros del grupo *Pendón sur e IR* que me han acompañado durante estos años, y a todas las personas que me han ayudado a probar la aplicación, con especial mención a Alejandro por su paciencia y ayuda.

Por último, a toda mi familia que siempre han estado ahí preguntando, interesándose por el desarrollo y ayudando en la medida de lo posible.

Resumen

OCAM – Outdoor Colective Activities Monitorization pretende reforzar la seguridad en actividades colectivas al aire libre. Este tipo de actividades, cuando son organizadas por asociaciones que siguen un Sistema de Gestión de Riesgos, precisan de un coordinador externo cuya misión es servir de enlace entre la actividad colectiva, a menudo mal comunicada por las condiciones orográficas y de cobertura de las redes móviles, y otros agentes externos (servicios de rescate, familias, transportes, etc.). Con la ayuda de las nuevas tecnologías se propone un sistema de rastro y monitorización de la posición del grupo que realiza la actividad.

OCAM cubre este hueco, permitiendo la gestión y monitorización de las actividades tanto desde dentro siendo participante, como desde fuera para los organizadores.

Esta gestión consiste el registro de actividades, participantes y constante monitorización de posición de todas las personas implicadas en las actividades en tiempo real, salvando los problemas de cobertura y ayudando a los guías de la actividad y coordinador externo en la gestión de las posibles vicisitudes que pueden ocurrir durante el desarrollo de la actividad.

Palabras Clave

Track, participante, guía, actividad, reporte, trayectoria.

Abstract

OCAM - Outdoor Colective Activities Monitorization aims to bolster security in outdoor group activities. This kind of activities, when being organised by associations that heed a Risk Management System, need an external coordinator whose task lies in serving as a link between the group activity, often suffering from a lack of communication due to its orographic condition and feeble coverage from mobile networks, and other external agents (rescue services, families, transports...). Therefore, it is suggested, by making use of new technologies, a trail and monitoring system of the group's doing the activity position.

OCAM fills this gap, allowing the management and monitoring of the activities, both from the inside being participant or from the outside for the organizers.

This management consists of the registration of activities, participants and continual monitorization of the position of all the people involved in the real-time activities, dealing with the problems related to coverage and supporting the guides of the activity and the external coordinator in the administration of possible vicissitudes that may occur during the activity.

Keywords

Track, hiker, guide, activity, report, path.

Índice General

CAPÍTULO 1. INTRODUCCIÓN.....	24
1.1 JUSTIFICACIÓN DEL PROYECTO	24
1.2 OBJETIVOS DEL PROYECTO	26
1.2.1 <i>Evaluación de alternativas</i>	27
1.3 SOLUCIÓN PROPUESTA.....	32
1.3.1 <i>Creación, gestión y monitorización de las actividades</i>	32
1.3.2 <i>Solución a falta de cobertura</i>	33
1.3.3 <i>Comparativa con alternativas</i>	33
CAPÍTULO 2. ASPECTOS TEÓRICOS.....	35
2.1 GPS.....	35
2.2 WAYPOINT	35
2.3 TRACK	36
2.3.1 <i>Ficheros GPX para tracks</i>	36
CAPÍTULO 3. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTO	39
3.1 PLANIFICACIÓN.....	39
3.1.1 <i>Planificación inicial</i>	39
3.1.2 <i>Planificación inicial del proyecto</i>	39
3.1.3 <i>Segunda planificación</i>	46
3.1.4 <i>Diario de trabajo</i>	46
3.2 RESUMEN DEL PRESUPUESTO	48
3.2.1 <i>Coste de recursos de trabajo</i>	48
3.2.2 <i>Recursos materiales</i>	48
3.2.3 <i>Costes por fases</i>	49
3.2.4 <i>Presupuesto interno</i>	49
3.2.5 <i>Presupuesto de cliente</i>	49
CAPÍTULO 4. ANÁLISIS	51
4.1 DEFINICIÓN DEL SISTEMA	51
4.1.1 <i>Determinación del Alcance del Sistema</i>	51
4.1.2 <i>Descripción de los subsistemas</i>	51
4.2 REQUISITOS DEL SISTEMA	53
4.2.1 <i>Obtención de los Requisitos del Sistema</i>	53
4.3 MODELO DE DOMINIO	60
4.4 ANÁLISIS DE CASOS DE USO Y ESCENARIOS.....	62
4.4.1 <i>Identificación de actores del sistema</i>	62
4.4.2 <i>Especificación de casos de uso y escenarios subsistema aplicación web</i>	62
4.4.3 <i>Especificación de casos de uso y escenarios subsistema aplicación móvil</i>	81
4.4.4 <i>Matriz de trazabilidad requisitos funcionales - casos de uso subsistema web</i>	97
4.4.5 <i>Matriz de trazabilidad requisitos funcionales – casos de uso subsistema aplicación móvil</i> . 98	98
4.5 ANÁLISIS DE INTERFACES DE USUARIO.....	99
4.5.1 <i>Primera versión de prototipos</i>	99
4.5.2 <i>Segunda versión de prototipos</i>	112
4.5.3 <i>Tercera versión de prototipos</i>	117
4.5.4 <i>Mapa de navegación definitivo</i>	125

4.5.5	<i>Matriz de trazabilidad interfaces - casos de uso subsistema web</i>	127
4.5.6	<i>Matriz de trazabilidad interfaces – casos de uso subsistema aplicación móvil</i>	127
4.6	ESPECIFICACIÓN DEL PLAN DE PRUEBAS	129
4.6.1	<i>Subsistema página web</i>	129
4.6.2	<i>Subsistema aplicación móvil</i>	134
CAPÍTULO 5.	DISEÑO DEL SISTEMA.....	141
5.1	ESPECIFICACIÓN Y JUSTIFICACIÓN DE LA PLATAFORMA TECNOLÓGICA UTILIZADA	141
5.1.1	<i>Spring Framework</i>	141
5.1.2	<i>Sistema Gestor de Base de Datos utilizado</i>	142
5.1.3	<i>Apache Tomcat</i>	142
5.1.4	<i>Android.....</i>	142
5.1.5	<i>Javascript</i>	144
5.1.6	<i>Angular JS.....</i>	144
5.2	ARQUITECTURA DEL SISTEMA	145
5.2.1	<i>Patrones de diseño utilizados.....</i>	145
5.2.2	<i>Diagrama de despliegue</i>	152
5.3	AUTENTICACIÓN DE USUARIOS	153
5.3.1	<i>JSON Web Tokens.....</i>	153
5.3.2	<i>Mantenimiento de la sesión</i>	155
5.4	DIAGRAMAS DE INTERACCIÓN DE CASOS DE USO	157
5.4.1	<i>Identificación en el sistema.....</i>	157
5.4.2	<i>Ver actividades.....</i>	157
5.4.3	<i>Iniciar actividad.....</i>	158
5.4.4	<i>Abandonar actividad.....</i>	159
5.4.5	<i>Unirse a actividad</i>	161
5.4.6	<i>Monitorizar actividad.....</i>	162
5.4.7	<i>Vincular cuenta</i>	163
5.5	DISEÑO DE LA BASE DE DATOS	164
5.5.1	<i>Descripción del SGDB usado.....</i>	164
CAPÍTULO 6.	IMPLEMENTACIÓN DEL SISTEMA.....	167
6.1	ESTÁNDARES Y NORMAS SEGUIDOS	167
6.1.1	<i>Java Code Conventions.....</i>	167
6.1.2	<i>Android Code Style and Guidelines</i>	167
6.1.3	<i>Usabilidad móvil y web: Material Design.....</i>	167
6.2	LENGUAJES DE PROGRAMACIÓN	169
6.2.1	<i>Java</i>	169
6.2.2	<i>Javascript</i>	169
6.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	170
6.3.1	<i>Spring Tool Suite</i>	170
6.3.2	<i>Android Studio</i>	170
6.3.3	<i>Atom</i>	171
6.3.4	<i>PgAdmin 4.....</i>	171
6.3.5	<i>Visual Paradigm</i>	172
6.3.6	<i>Turbo.net y Browsershots</i>	172
6.4	LIBRERÍAS USADAS PARA EL DESARROLLO	173
6.4.1	<i>Angular 1.....</i>	173
6.4.2	<i>Angular Material.....</i>	173
6.4.3	<i>Angular Google Maps</i>	174
6.4.4	<i>OSMDroid.....</i>	174

6.4.5	<i>Fabric</i>	174
6.4.6	<i>Volley</i>	175
6.4.7	<i>GreenRobot: greenDAO</i>	175
6.4.8	<i>Gson</i>	175
6.4.9	<i>JavaMail API</i>	175
6.4.10	<i>Jasmine</i>	176
6.4.11	<i>JUnit</i>	176
6.5	PROBLEMAS ENCONTRADOS	177
6.5.1	<i>Consumo de batería</i>	177
6.5.2	<i>Precisión de los reportes</i>	179
6.5.3	<i>Requerimiento de permisos</i>	180
6.5.4	<i>Almacenamiento de datos para funcionamiento offline</i>	180
6.5.5	<i>Ejecución de tareas en segundo plano</i>	180
6.6	DESCRIPCIÓN DETALLADA DEL SISTEMA DESARROLLADO	182
6.6.1	<i>Diagrama de paquetes servidor</i>	182
6.6.2	<i>Diagrama de paquetes aplicación móvil</i>	184
6.6.3	<i>Descripción elementos aplicación web</i>	186
6.6.4	<i>Descripción detallada de las clases</i>	190
CAPÍTULO 7. DESARROLLO DE LAS PRUEBAS		196
7.1	PRUEBAS UNITARIAS	196
7.2	PRUEBAS DE COMPORTAMIENTO.....	197
7.3	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA.....	199
7.3.1	<i>Subsistema página web</i>	199
7.3.2	<i>Subsistema aplicación móvil</i>	202
7.4	PRUEBAS DE USABILIDAD	207
7.4.1	<i>Subsistema aplicación web</i>	207
7.4.2	<i>Subsistema aplicación móvil</i>	209
7.5	PRUEBAS DE ACCESIBILIDAD.....	211
7.5.1	<i>Subsistema aplicación web</i>	211
7.5.2	<i>Subsistema aplicación móvil</i>	217
7.6	PRUEBAS DE RENDIMIENTO	219
7.6.1	<i>Primera prueba</i>	219
7.6.2	<i>Segunda prueba</i>	220
7.6.3	<i>Tercera prueba</i>	221
7.6.4	<i>Conclusiones</i>	222
CAPÍTULO 8. MANUALES DEL SISTEMA		225
8.1	MANUAL DE INSTALACIÓN	225
8.1.1	<i>Instalación del servidor</i>	225
8.1.2	<i>Aplicación móvil</i>	226
8.1.3	<i>Aplicación web</i>	226
8.2	MANUAL DE EJECUCIÓN	227
8.3	MANUAL DE USUARIO	229
8.3.1	<i>Aplicación web</i>	229
8.3.2	<i>Aplicación móvil</i>	239
8.4	MANUAL DEL PROGRAMADOR.....	249
8.4.1	<i>Servidor</i>	249
8.4.2	<i>Aplicación móvil</i>	249
8.4.3	<i>Aplicación web</i>	250
CAPÍTULO 9. CONCLUSIONES Y AMPLIACIONES		252

9.1	CONCLUSIONES.....	252
9.1.1	<i>Conclusiones técnicas.....</i>	252
9.1.2	<i>Conclusiones académicas.....</i>	253
9.1.3	<i>Conclusiones personales</i>	253
9.2	AMPLIACIONES	255
CAPÍTULO 10. REFERENCIAS BIBLIOGRÁFICAS.....		257
CAPÍTULO 11. APÉNDICES		259
11.1	DIARIO DE TRABAJO	259
11.2	CONTENIDOS ENTREGADOS	267
11.2.1	<i>Estructura de la carpeta entregada.....</i>	267
11.2.2	<i>Estructura del código fuente.....</i>	268

Índice de Figuras

Figura 1 Logo FEMPA.....	24
Figura 2 Logo MIDE	24
Figura 3 Captura de pantalla Alpify	27
Figura 4 Esquema de funcionamiento de locateU	28
Figura 5 Capturas de pantalla aplicación SafeMountain	29
Figura 6 Capturas de pantalla aplicación Mountain Bike SOS	30
Figura 7 Capturas de pantalla aplicación SOS Sports	30
Figura 8 Esquema de funcionamiento de la aplicación a alto nivel	33
Figura 9 Estudios iniciales y formación de la planificación inicial.....	40
Figura 10 Gantt fases de estudios iniciales y formación	40
Figura 11 Fase de análisis y diseño planificación inicial	40
Figura 12 Gantt fase de análisis y diseño planificación inicial	41
Figura 13 Fase de implementación planificación inicial	41
Figura 14 fase de pruebas y documentación planificación inicial.....	41
Figura 15 Gantt fases pruebas y documentación planificación inicial	42
Figura 16 Gantt marzo - mayo planificación inicial	43
Figura 17 Gantt enero- marzo planificación inicial.....	44
Figura 18 Gantt octubre-enero planificación inicial	45
Figura 19 Gráfico circular comparación de horas por fases del proyecto	47
Figura 20 Esquema de funcionamiento subsistemas	52
Figura 21 Modelo de dominio	60
Figura 22 Diagrama casos de uso subsistema aplicación web.....	62
Figura 23 Diagrama de actividad - CU-W-1 Registrarse.....	64
Figura 24 Diagrama de actividad - CU-W-2 Iniciar sesión.....	66
Figura 25 Diagrama de actividad - CU-W-3 Ver actividades	68
Figura 26 Diagrama de actividad - CU-W-4 Recuperar cuenta de usuario	69
Figura 27 Diagrama de actividad – CU-W-5 cerrar sesión	70
Figura 28 Diagrama de actividad - CU-W-6 Borrar cuenta.....	71
Figura 29 Diagrama de actividad - CU-W-7 Crear actividad.....	73
Figura 30 Diagrama de actividad - CU-W-8 Editar actividad.....	75
Figura 31 Diagrama de actividad – CU-W-10 Editar datos personales.....	76
Figura 32 Diagrama de actividad - CU-W-10 Monitorizar actividad	78
Figura 33 Eliminar actividad	80
Figura 34 Diagrama de casos de uso subsistema aplicación móvil	81
Figura 35 Diagrama de actividad - CU-M-1 Registrarse.....	83
Figura 36 Diagrama de actividad – CU-M-2 Vincular cuenta	84
Figura 37 Diagrama de actividad - CU-M-3 Ver actividades	85
Figura 38 Diagrama de actividad – CU-M-4 Comenzar actividad.....	87
Figura 39 Diagrama de actividad – CU-M-5 Monitorizar actividad aplicación móvil	88
Figura 40 Diagrama de actividad – CU-M-6 Unirse a actividad	90
Figura 41 Diagrama de actividad – CU-M-7 Finalizar actividad	91
Figura 42 Diagrama de actividad - CU-M-8 Cambiar password	92
Figura 43 Diagrama de actividad - Abandonar actividad	94
Figura 44 Diagrama de actividad - Cerrar sesión.....	95
Figura 45 Prototipo de login - Primera versión	100
Figura 46 Prototipo de registro - Primera versión.....	101

Figura 47 Prototipo página principal - Primera versión	101
Figura 48 Prototipo Datos Personales - Primera versión	102
Figura 49 Prototipo nueva actividad - Primera versión	103
Figura 50 Prototipo track de la ruta - Primera versión	104
Figura 51 Prototipo vista de actividad - Primera versión.....	105
Figura 52 Prototipo editar actividad - Primera versión	106
Figura 53 Prototipo monitorizar actividad - Primera versión	107
Figura 54 Prototipo acceso excursionista - Primera versión	108
Figura 55 Prototipo acceso guía - Primera versión	109
Figura 56 Prototipo Actividades – Primera versión	110
Figura 57 Prototipo actividad - Primera versión.....	111
Figura 58 Prototipo monitorización – Primera versión.....	111
Figura 59 Prototipo monitorización excursionistas – primera versión.....	112
Figura 60 Prototipo (v2) página principal	113
Figura 61 Prototipo (v2) nueva actividad	114
Figura 62 Prototipo selección de guías.....	115
Figura 63 Prototipo (v2) monitorización actividad	115
Figura 64 Prototipo (v2) monitorización sobre mapa	116
Figura 65 Prototipo (v2) Vinculación de la cuenta.....	116
Figura 66 Prototipo (v2) Actividad	117
Figura 67 Prototipo (v3) login	118
Figura 68 Prototipo (v3) actividades	119
Figura 69 Prototipo (v3) nueva actividad	120
Figura 70 Prototipo (v3) monitorización	121
Figura 71 Prototipo (v3) registro.....	122
Figura 72 Prototipo (v3) Actividades.....	122
Figura 73 Prototipo (v3) actividad.....	123
Figura 74 Prototipo (v3) monitorización	124
Figura 75 Prototipo perfil.....	124
Figura 76 Mapa de navegación – subsistema aplicación web.....	125
Figura 77 Mapa de navegación – aplicación móvil	126
Figura 78 Logo Spring Framework.....	141
Figura 79 Logo Framework Ionic	143
Figura 80 Logo Angular JS	144
Figura 81 Esquema modelo de Brown N-Capas.....	145
Figura 82 Estructura patrón Modelo Vista Presentador (MVP)	147
Figura 83 Diagrama de clases Activity Configuración – aplicación Android	148
Figura 84 Estructura patrón strategy	148
Figura 85 Estructura patrón Singleton	149
Figura 86 Diagrama de clases gestión de tipos de mapas – Aplicación Android	150
Figura 87 Estructura patrón state	151
Figura 88 diagrama de clases envío de reportes – Aplicación Android	152
Figura 89 Diagrama de despliegue	152
Figura 90 Esquema de funcionamiento de Autenticación por token	154
Figura 91 Esquema de funcionamiento de JWT con varios clientes	155
Figura 92 Diagrama de interacción – Autenticación servidor	157
Figura 93 Diagrama de interacción - Ver actividades	158
Figura 94 Diagrama de interacción – Iniciar actividad	159
Figura 95 Diagrama de interacción – Abandonar actividad	159
Figura 96 Diagrama de interacción – Unirse actividad	161
Figura 97 Diagrama de interacción – Monitorizar actividad	162

Figura 98 Diagrama de interacción – Vincular cuenta.....	163
Figura 99 Diagrama Base de datos	164
Figura 100 Versión de uso del IDE Spring Tool Suite	170
Figura 101 Versión del uso del IDE Android Studio	171
Figura 102 Versión de uso de Atom.....	171
Figura 103 Versión de uso de PgAdmin 4.....	172
Figura 104 Versión de uso herramienta Visual Paradigm	172
Figura 105 Logo AngularJS.....	173
Figura 106 Logo Angular Material	173
Figura 107 Estado de OCAM – móvil en Fabric	174
Figura 108 Logo Jasmine	176
Figura 109 Logo JUnit	176
Figura 110 Consumo de batería aplicación móvil	177
Figura 111 Desglose de consumo de batería OCAM	178
Figura 112 Desglose consumo de batería aplicación análoga	178
Figura 113 Resumen de actividad sobre el mapa en actividad análoga	179
Figura 114 Resumen de actividad en OCAM.....	179
Figura 115 Diagrama de paquetes servidor com.ocam.service	182
Figura 116 Diagrama de paquetes servidor com.ocam.model	182
Figura 117 Diagrama de paquetes servidor com.ocam.util, com.ocam.repository	183
Figura 118 Diagrama de paquetes servidor com.ocam.ws.....	183
Figura 119 Diagrama de paquetes servidor	184
Figura 120 Diagrama de paquetes aplicación móvil.....	185
Figura 121 Pruebas JUnit servidor	196
Figura 122 Pruebas Jasmine aplicación web	197
Figura 123 Gráfico de barras media de notas preguntas formulario usabilidad	208
Figura 124 Aplicación web en chrome versión 50.0.266	211
Figura 125 Chrome versión 58.0.3029	212
Figura 126 Aplicación web en Firefox 22.0	212
Figura 127 Aplicación web en Firefox 53.0.3	212
Figura 128 Aplicación web en Opera 33.0.....	212
Figura 129 Aplicación web en Opera 35.0.....	213
Figura 130 Aplicación web en Safari 10	213
Figura 131 Aplicación web en SeaMonkey 2.22.1	213
Figura 132 Aplicación web en SeaMonkey	214
Figura 133 Lista de actividades con alta resolución	214
Figura 134 Listado de actividades con resolución 412x732	215
Figura 135 Listado de actividades con resolución 732x412	215
Figura 136 Vista de monitorización con resolución 1920x1080.....	216
Figura 137 Monitorización en resolución 412x732	216
Figura 138 Listado de actividades móvil en 1440x2560dpi	217
Figura 139 Listado de actividades móvil en 2560x1600dpi	218
Figura 140 Grupo de usuarios primera prueba rendimiento – Jmeter	219
Figura 141 Petición HTTP realizada primera prueba rendimiento JMeter.....	219
Figura 142 Resultados primera prueba rendimiento – Jmeter	220
Figura 143 Grupo de usuarios segunda prueba rendimiento JMeter	220
Figura 144 Resultados segunda prueba rendimiento JMeter.....	220
Figura 145 Resultados tercera prueba rendimiento JMeter.....	221
Figura 146 Latencia de las peticiones tercera prueba rendimiento JMeter.....	221
Figura 147 Códigos de respuesta HTTP peticiones prueba 3 rendimiento JMeter.....	222
Figura 148 Instalación de aplicaciones de orígenes desconocidos	226

Figura 149 Ejecución del servidor desde entorno de desarrollo	227
Figura 150 Exportación como fichero war paso 1.....	227
Figura 151 Exportación como fichero war – Paso 2	228
Figura 152 Login aplicación web	229
Figura 153 Formulario de registro.....	230
Figura 154 Aviso de envío de email de confirmación	230
Figura 155 Email de confirmación de cuenta	230
Figura 156 Aviso de cuenta validada.....	231
Figura 157 Listado de actividades pendientes página web.....	231
Figura 158 Detalle de una actividad en curso.....	232
Figura 159 Track de la ruta.....	232
Figura 160 Password para monitorizar actividad	233
Figura 161 Excursionistas monitorizados vista de monitorización.....	233
Figura 162 Monitorización de un excursionista	234
Figura 163 Actividades realizadas	235
Figura 164 Detalle de actividad concluida.....	235
Figura 165 Resumen de monitorización	236
Figura 166 Creación de nueva actividad.....	237
Figura 167 Nuevo guía en creación de actividad	237
Figura 168 Guardar actividad	238
Figura 169 Confirmación de eliminación de actividad	238
Figura 170 Mi cuenta	239
Figura 171 Cambiar password.....	239
Figura 172 Login aplicación móvil	240
Figura 173 Registro aplicación móvil.....	240
Figura 174 Listado actividades sin conexión aplicación móvil	240
Figura 175 Filtrado lista actividades aplicación móvil	241
Figura 176 Recarga lista actividades aplicación móvil	241
Figura 177 Detalle de una actividad – aplicación móvil.....	241
Figura 178 Solicitud de permisos aplicación android	242
Figura 179 Tipos de mapa aplicación android	242
Figura 180 Introducción password para comenzar actividad	243
Figura 181 Cambiar password actividad	243
Figura 182 Password actividad unirse	244
Figura 183 Monitorización actividad aplicación móvil	244
Figura 184 Posición excursionista mapa.....	245
Figura 185 Notificación participante actividad aplicación móvil	245
Figura 186 Confirmación abandono de actividad aplicación móvil.....	246
Figura 187 Menú de navegación aplicación móvil.....	247
Figura 188 Configuración temporalidad reportes aplicación móvil	247
Figura 189 Java 1.8 compiler.....	249
Figura 190 Android SDK Manager	249
Figura 191 Arrancando browser-sync en localhost:3000	250
Figura 192 Estructura carpeta OCAM – src 1	268
Figura 193 Estructura carpeta OCAM – src 2	269
Figura 194 Estructura carpeta OCAM-web – src common	270
Figura 195 Estructura carpeta OCAM-web – src private	271
Figura 196 Estructura carpeta OCAM-web – src public	272
Figura 197 Estructura carpeta OCAM-android – src 1.....	273
Figura 198 Estructura carpeta OCAM-android – src 2.....	274
Figura 199 Estructura carpeta OCAM-android – src 3.....	275

Índice de tablas

Tabla 1 Comparativa de aplicación propuesta con alternativas	33
Tabla 2 Desglose de porcentajes de horas en planificación inicial.....	39
Tabla 3 Resumen de horas.....	47
Tabla 4 Recursos de trabajo - presupuesto	48
Tabla 5 Recursos materiales - presupuesto.....	48
Tabla 6 Coste de energía - presupuesto	49
Tabla 7 Costes por fases presupuesto	49
Tabla 8 Presupuesto interno.....	49
Tabla 9 Presupuesto de cliente	49
Tabla 10 Identificación de actores del sistema.....	62
Tabla 11 Casos de uso subsistema aplicación web	63
Tabla 12 Matriz de trazabilidad requisitos funcionales con casos de uso – subsistema web	97
Tabla 13 Matriz de trazabilidad requisitos funcionales con casos de uso - subsistema móvil.....	98
Tabla 14 Matriz de trazabilidad interfaces con casos de uso subsistema web.....	127
Tabla 15 Matriz de trazabilidad interfaces con casos de uso subsistema aplicación móvil	127
Tabla 16 Comparativa float – double Java	179
Tabla 17 Notas medias preguntas formulario usabilidad	208
Tabla 18 Sugerencias usabilidad subsistema web	209
Tabla 19 Sugerencias usabilidad subsistema aplicación móvil	210
Tabla 20 Versiones de navegadores – aplicación web.....	214
Tabla 21 Resumen pruebas rendimiento	222
Tabla 22 Diario de trabajo	266

Capítulo 1. Introducción

1.1 Justificación del proyecto

Tanto a nivel comunitario como nacional, hoy en día se están empleando muchos recursos y tiempo en reducir la accidentalidad durante actividades de montaña, y actividades al aire libre en general. La accidentalidad en este tipo de actividades que implican desplazamientos, no solo supone un riesgo de salud, sino que también genera una responsabilidad jurídica debido a los rescates que conllevan dichos accidentes. Con el fin de fomentar el conocimiento previo sobre una ruta, se diseñó el Método de Información De Excursiones (MIDE). Éste, pretende unificar la forma en la que se expresa la dificultad y características de un recorrido y se diseñó como herramienta de prevención de accidentes. Su uso es libre, y con él se puede crear un informe completo sobre los detalles de una ruta, condiciones, itinerario, desplazamiento, dificultad, etc.

Así mismo, del interés de optimizar la organización de las actividades colectivas, y centrado en aspectos de seguridad física, aparece el Sistema de Gestión de Riesgos (SGR), desarrollado y promovido por la Federación de Deportes de Montaña, Escalada y Senderismo del Principado de Asturias (FEMPA). La finalidad de dicho sistema es incrementar la seguridad física y desarrollar la seguridad jurídica para así, entre otras cosas, disminuir los accidentes de montaña.



Figura 1 Logo FEMPA



Figura 2 Logo MIDE

Durante una actividad al aire libre, por ejemplo, de montaña, existen muchos riesgos que han de controlarse y prevenirse. Actualmente, en aquellos grupos que hayan implantado un SGR para sus actividades, cuando se va a realizar una actividad, el guía de la misma elabora un informe en el que se incluye, entre otras cosas, la ruta que se va a seguir, por donde va a pasar, la posición de inicio y fin, la dificultad, etc. Dicho informe, que además es recomendable que se haya elaborado mediante el MIDE, se le hace llegar a todos los asistentes a la actividad previamente a ella, para que sean conscientes de todos los detalles.

En aquellas actividades en las que los organizadores apliquen un SGR, están siempre gestionadas por un mínimo de dos guías además de los participantes, los cuales se sitúan uno al principio y otro al final de la expedición. De esta forma, cualquier cosa que pase entre cualquiera de los dos puntos está controlada y supervisada. Por otro lado, estas dos personas tienen la vital responsabilidad de mandar reportes periódicos a un coordinador externo de la

actividad. Éste no asiste a la misma, y su misión es realizar un seguimiento continuo de cómo se está realizando y si todo transcurre con total normalidad. En caso que ocurra algún problema o incidente, con el correspondiente reporte al coordinador externo, se pueden llevar a cabo los trámites necesarios para solventarlo, como puede ser, por ejemplo, gestionar un rescate, mandar ayuda... etc.

Asimismo, cualquier alteración del plan inicial especificado en el informe, que se decida sobre la marcha de la actividad, debe ser también notificada al coordinador externo. Por ejemplo, que en un momento dado se vea la necesidad de cambiar el punto de llegada por imposibilitarse el paso en determinado lugar. Esto debe ser notificado al coordinador externo, el cual tendrá la responsabilidad de avisar o realizar los cambios pertinentes.

De la misma forma, cualquier otro elemento que intervenga en la actividad, por ejemplo, transporte, también tiene que estar comunicado y disponible de continuo. En resumen, el coordinador externo tiene que estar al corriente de todos los elementos que intervengan durante la actividad y del transcurso de la misma, para así poder responder de manera rápida y eficiente ante cualquier imprevisto. Cualquier carencia de información, en un momento dado, puede ocasionar que no pueda llevar a cabo su tarea.

La mecánica ideal de este sistema es que tanto los dos guías que acompañan la expedición como todos los demás componentes que intervengan en ella, sean capaces de mantener la dinámica de envío de reportes con la información del progreso de la actividad de forma ininterrumpida. La realidad es que esto no siempre se puede llevar a cabo.

Uno de los problemas más comunes por el cual no se mandan reportes al coordinador externo de la actividad con la suficiente frecuencia, es la falta de cobertura en determinados puntos de la actividad. Si en un momento dado, se decide desviar la ruta por un camino alternativo al original y en ese momento no es posible enviarlo, esa información ya no va a llegar al coordinador externo y tendría que estar al tanto de ello.

Otro de los problemas más frecuentes en este tipo de casos es el olvido de enviar información. Para las personas que llevan la actividad es difícil mantener un ritmo periódico de envío de información y muchas veces puede ocurrir que este proceso se pase por alto.

Por todo ello, existe un problema evidente de que no hay ningún sistema en la actualidad que nos garantice la entrega de la información de forma periódica y completa durante el transcurso de la actividad, para que en todo momento se sepa por donde va exactamente la expedición, de cuantas personas está formada y si se está siguiendo el transcurso normal esperado para la misma.

1.2 Objetivos del proyecto

La meta del proyecto es **incrementar la seguridad en actividades colectivas al aire libre**. Esto se podría desglosar en los siguientes principales objetivos:

- Creación y configuración de actividades
- Gestión de actividades y participantes
- Monitorización de las actividades y sus participantes en tiempo real

La **creación y la configuración de las actividades** tiene por objetivo el poder darlas de alta en la aplicación, que sean visibles por cualquier otro usuario y, por medio de los campos que la aplicación recogerá de las mismas, permitir llegar a un alto nivel de detalle en cada una de ellas.

La **gestión de las actividades** tiene por objetivo el permitir tener un ciclo de vida para cada una de ellas, desde que se crean hasta que se cierran, en el que intervengan otros usuarios, ya sea como guías, participantes o simplemente coordinadores.

El objetivo de esto es dar una funcionalidad totalmente versátil con el fin de poder gestionar a través de la aplicación cualquier cambio o acontecimiento con la actividad real. De esta forma, a través de la propia aplicación se puede llevar un seguimiento real de la actividad.

El objetivo de la **monitorización** está orientado a poder conocer, ya seas una persona externa a la actividad que no participe o bien un participante, el transcurso de la misma. Esto permitirá conocer el estado de posición en cada momento de cada persona participante en la actividad.

La información relativa a los participantes será enviada de forma periódica y constante durante el transcurso de la actividad, con el objetivo de poder llevar un seguimiento real del transcurso de la misma.

Todo lo anterior contribuye a **incrementar la seguridad** en aplicaciones al aire libre. No está entre sus objetivos principales actuar como red social para compartir actividades, sino como una herramienta para prevenir accidentes y para llevar mejores controles de todos los aspectos susceptibles de ser monitorizados en una actividad al aire libre.

1.2.1 Evaluación de alternativas

En la actualidad existen algunas aplicaciones móviles que ayudan a solventar algunos de los problemas anteriormente descritos. En muchas ocasiones no son suficientemente útiles como para poder depender de ellas en su totalidad como transmisoras de la información del transcurso de la actividad o como gestoras de personas en actividades al aire libre. En este apartado veremos y analizaremos las más destacadas hoy en día para posteriormente compararlas con la propuesta.

1.2.1.1 Alpify

El principal objetivo de esta aplicación y que lleva a cabo es la comunicación del excursionista con los servicios de emergencia. Actualmente funciona en todo el territorio nacional. Esta aplicación entra en funcionamiento una vez ha ocurrido un problema, es decir, si una actividad transcurre con total normalidad, Alpify no es útil.

Una vez ha ocurrido algo, se ha perdido o accidentado una persona, ella misma puede activar la aplicación la cual mandará un reporte de situación a los servicios de rescate. En este caso no solventa los problemas planteados anteriormente ya que como hemos mencionado, Alpify solamente entra en juego una vez que algo ha ido mal.

Por otro lado, Alpify solo hace comunicación con los servicios de emergencia, esto significa que solo ellos pueden obtener la información del accidentado y nadie más puede saber nada acerca del transcurso de la actividad. Además, solo se tiene en cuenta a la persona que manda el reporte de accidente, en ningún momento se sabe si va acompañado o no, o la situación de las demás personas.



Figura 3 Captura de pantalla Alpify

1.2.1.2 *locateU*

La aplicación locateU es una herramienta de seguridad para cualquier actividad al aire libre que implique una posición de partida y una de llegada.



Figura 4 Esquema de funcionamiento de locateU

Antes de realizar una actividad deberás configurar la aplicación: establecer una posición de partida y otra de llegada, un tiempo de espera máximo para la situación de emergencia y un contacto por si pasase algo, entre otras posibles configuraciones. Una vez que la actividad haya comenzado la aplicación irá recogiendo datos de forma interna sobre la posición de la persona. Si en un momento dado la persona se queda estática en un mismo punto durante mucho tiempo, esto se interpreta como que el excursionista ha tenido algún problema. Pasado el tiempo de espera configurada donde puedes cancelar dicha emergencia, se dará una señal de alarma y se avisará por correo electrónico al contacto de emergencia con los datos de posición de la persona.

Pensando en el problema que queremos solucionar, esta aplicación no abarca todas las funcionalidades necesarias. En primer lugar, está pensada para un solo excursionista, es decir, con ella no puedes organizar grupos. Además, de la misma forma que pasaba con Alpify, en ningún momento nadie que no esté en la actividad puede llevar un seguimiento continuo sobre el desarrollo de la misma.

1.2.1.3 *Safemountain*

Con Safemountain lo primero que puedes hacer es configurar la ruta que vas a seguir. Permite llegar a un alto nivel de detalle incluyendo incluso desniveles, dificultad del terreno, distancia, tiempo, etc. Esta aplicación necesita de conexión GPS para funcionar, aunque también lo hace sin cobertura gracias al sistema de localización llevado a cabo por ellos mismos.

Safemountain implementa un sistema de notificaciones que puede ser de dos tipos. Notificaciones generadas por terceros, esto es, noticias sobre espacios rurales para los cuales tienes programada una actividad, y notificaciones propias. Éstas van asociadas a tus recorridos activos y permiten generar llamadas de auxilio en caso de emergencia. Además, funciona también como una red social pudiendo añadir contactos para compartir información a través de la aplicación u otras redes sociales.



Figura 5 Capturas de pantalla aplicación SafeMountain

A pesar de que la aplicación tiene funcionalidad de seguridad en la montaña, que es lo que realmente estamos buscando, la orientación general es más cara a red social de contactos con los que transmitir información relacionada con las rutas. Por tanto, carece de funcionalidad como la de gestionar grupos de personas, y más importante, de envío continuo de información de la posición GPS.

1.2.1.4 *Mountain Bike SOS*

Aunque está pensada para actividades en bicicleta, está recomendada también en algunos foros de montañeros para su uso durante actividades al aire libre en general. Primero que nada, deberás configurar un contacto de emergencia. Su única función es la de localizar a un accidentado y funciona con dos modos de alerta, manual y automático.

Con el modo manual activo es la propia persona accidentada la que tiene que dar la señal de SOS desde la aplicación, es decir, tiene que estar consciente para ello. Con el modo automático la aplicación registrará las llamadas entrantes al teléfono móvil. Si se reciben un número configurado previamente de llamadas no contestadas del contacto de emergencia durante el transcurso de la actividad, la aplicación mandará automáticamente un SMS de alerta a dicho contacto con los datos de la posición del dispositivo. Necesitará cobertura en todo momento para funcionar, ya que su única funcionalidad automática está limitada a poder recibir llamadas.



Figura 6 Capturas de pantalla aplicación Mountain Bike SOS

Con respecto a nuestro problema, es una aplicación que puede ser útil si hay una persona que nos llame de forma frecuente. Por otro lado, no es práctica, es decir, para que alguien sepa dónde está el excursionista tendría que estar continuamente llamando a su contacto. Esto no nos libera del problema de mandar reportes de forma continua, ya que se convierte en eso mismo. Además, está pensada para actividades que se realizan de forma individual y no en grupo.

1.2.1.5 SOS Sports

Esta aplicación está pensada para personas que realizan una actividad al aire libre de manera individual. Funciona de forma muy similar a locateU. La aplicación supervisa de forma continua tu posición GPS y si durante un tiempo previamente configurado no detecta movimiento mandará un SMS al contacto que tengas predefinido con tus datos de posición actual.



Figura 7 Capturas de pantalla aplicación SOS Sports

Necesita de cobertura para funcionar y como vimos con locateU, no nos libera de los principales problemas de enviar información de nuestra posición de forma continua ya que solo funcionará cuando ocurra una emergencia.

1.3 Solución propuesta

La solución que planteamos a este problema es realizar una aplicación que abarque todas las necesidades mencionadas anteriormente para una excursión de montaña. Constará de una aplicación web donde poder ver todos los datos de actividades, registrarlas, etc, de una aplicación nativa Android, la cual extraerá toda la información del progreso de una actividad y de un servidor donde se procesarán todos los datos enviados tanto por la página web como la aplicación móvil.

En primer lugar, y la principal diferencia con las otras aplicaciones mencionadas anteriormente, es que permitirá gestionar grupos de excursionistas y no solo de manera individual para la persona que lleva un dispositivo móvil conectado con la aplicación.

1.3.1 Creación, gestión y monitorización de las actividades

Antes de realizar una actividad, la persona que realiza el informe de la misma, la configurará en la página web incluyendo todos los detalles. Durante la configuración de la actividad se deberá introducir toda la información relativa a la ruta que se seguirá: localización de comienzo y fin, horas esperadas de llegada a determinados puntos, fecha de la actividad, track de la ruta, guías que asistirán, etc. Una vez ha llegado el día y hora marcada para la actividad, todos los asistentes que quieran ser monitorizados podrán serlo solamente accediendo a ella a través de la aplicación móvil y una contraseña de acceso que proporcione uno de los guías de la actividad.

La aplicación móvil liberará a los expertos que asisten a la actividad de la responsabilidad de enviar reportes de posición continuos al coordinador externo, ya que lo hará en todo momento de forma automática para cada uno de los asistentes. Esta información será enviada desde cada dispositivo móvil y procesada por el servidor. Desde la aplicación web se podrá ver toda esta información de posición de excursionistas procesada. En el momento que se observe cualquier anomalía sobre el progreso esperado para la actividad, gracias a la información que se está recibiendo de forma continua, se podrán tomar las medidas necesarias de forma rápida y eficaz.

Además del propio coordinador externo, cualquier persona que esté físicamente en la actividad y siendo monitorizado, podrá también conocer la geolocalización del resto de asistentes a través de la aplicación móvil. Esto ayudará a que en caso de que algún excursionista se pierda, pueda reengancharse con el grupo, o, por otro lado, que el grupo sea capaz de encontrar a la persona perdida.



Figura 8 Esquema de funcionamiento de la aplicación a alto nivel

1.3.2 Solución a falta de cobertura

Ante el problema de la falta de cobertura, si la aplicación móvil en un momento dado no puede enviar la información, se encolará de forma local. En el momento que el dispositivo disponga de nuevo de conexión a internet todos los datos almacenados se enviarán juntos. De esta forma nos aseguramos que ninguna información es descartada. Además de reportes, la aplicación móvil almacenará de forma local acciones que se lleven a cabo sin conexión. Dichas acciones, con los parámetros introducidos por el usuario a cada una de ellas, serán procesadas también una vez recuperada la conexión a internet. Esto puede ser unirse a una actividad, darla por cerrada o comenzarla, abandonarla... etc.

Si se accede a la aplicación desde un primer momento sin conexión a internet, se mostrarán todos los datos almacenados de la última sesión que se realizó con conexión, pudiendo llevar a cabo casi la totalidad de las acciones que se pueden llevar sin la conectividad.

1.3.3 Comparativa con alternativas

A continuación, se muestra una comparativa de las principales aplicaciones para actividades al aire libre mencionadas anteriormente con la solución propuesta, con el fin de ver de forma más clara las diferencias entre unas y otras.

	Alpify	locateU	SafeMountain	Mountain Bike SOS	SOS Sports	Aplicación propuesta
Mapa de ruta	No	No	No	No	No	Sí
Localización GPS	Sí	Sí	Sí	Sí	Sí	Sí
Gestión de grupos	No	No	No	No	No	Sí
Funcionamiento sin conexión	Sí	No	Sí	No	No	Sí

Tabla 1 Comparativa de aplicación propuesta con alternativas

Como se observa en la Tabla 1, tenemos listadas por filas todas las funcionalidades deseadas para una aplicación de seguridad en montaña, que además gestione grupos. Habiendo estudiado las principales aplicaciones en el mercado en la actualidad, la propuesta de aplicación para personas al aire libre abarca todas las funcionalidades deseadas.

Asimismo, cabe destacar que ninguna de las aplicaciones encontradas como posibles alternativas a la propuesta soluciona el problema de la gestión de grupos. Todas están orientadas a usarse de manera individual como usuario único de la aplicación, sin ningún tipo de interacción con otros.

Capítulo 2. Aspectos Teóricos

En este capítulo se darán a conocer aquellos aspectos teóricos relevantes necesarios para comprender el desarrollo y funcionamiento del proyecto.

En OCAM, tanto para representar la trayectoria de posiciones GPS que se espera seguir durante la actividad, como para la representación de la trayectoria real que ha seguido, o está siguiendo un excursionista, se usan tracks.

Los tracks son utilizados, para entre otros, describir los waypoints que forman la ruta en cuestión. Dichos tracks, tanto en la aplicación móvil, como web y posteriormente, en el servidor, son interpretados y transferidos mediante ficheros GPX.

2.1 GPS

De las siglas Sistema de Posicionamiento Global (Global Positioning System), se trata de un sistema que permite identificar localizaciones sobre el espacio terrestre, sea donde sea que se encuentren.

Actualmente para su funcionamiento se usan 24 satélites en órbita sobre la Tierra, los cuales, entre todos, cubren toda la superficie. Cuando se quiere hacer uso de ellos el receptor de la posición localiza al menos tres de los 24 satélites.

El receptor recibe unas señales de cada uno de ellos en las que se encuentran la hora de reloj y la identificación. En base a esta información el receptor puede calcular el tiempo que han tardado las señales en llegar, y por tanto, también la distancia. Sabiendo también la posición exacta de cada uno de ellos, se obtiene finalmente la posición del propio receptor.

2.2 Waypoint

Un waypoint es un conjunto de coordenadas que permiten identificar un punto, una localización GPS de forma única y exacta. En la aplicación se usan para formar los tracks de la ruta a seguir y las trayectorias que siguen los excursionistas durante el transcurso de las actividades.

Para representar puntos terrestres, los utilizados en la aplicación, un waypoint está formado por dos coordenadas: la latitud y la longitud. Para otro tipo de aplicaciones, véase la navegación aérea, también se introduce la altitud a la información que conforma el waypoint.

En OCAM, aquellos waypoints que representan las posiciones por las que pasa un excursionista en el transcurso de una actividad son extraídos por la aplicación móvil de forma periódica en intervalos de tiempo establecidos por el propio usuario. Interpretados uno tras otro se conforma el track, la ruta que ha seguido ese excursionista en el espacio físico terrestre.

2.2.1.1 Estructura de un waypoint

Un waypoint, entre otros atributos dependiendo de la finalidad del mismo, puede contener la siguiente información:

- Nombre identificativo del waypoint
- Dirección escrita del lugar que representa el waypoint
- Latitud
- Longitud

Un ejemplo de waypoint:

Club Náutico de Ibiza

LATITUD: 38º 54' 06 N

LONGITUD: 01º 25' 06 E

2.3 Track

Un track se puede entender como una sucesión de puntos que conforman un recorrido, pudiendo ser representado sobre un mapa para apreciar, en el caso de OCAM, el recorrido esperado de una ruta o bien la ruta seguida por un excursionista durante el desarrollo de una actividad.

2.3.1 Ficheros GPX para tracks

La forma de almacenar tracks y transferirlos entre cliente y servidor en OCAM es mediante ficheros GPX. Se trata de un esquema XML estandarizado con contenido orientado a la transferencia de información de tracks, waypoints y otros datos de interés relacionados con el propio track en cuestión.

Un ejemplo de track puede ser el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.0">
  <name>Example gpx</name>
  <wpt lat="46.57638889" lon="8.89263889">
    <ele>2372</ele>
    <name>LAGORETICO</name>
  </wpt>
  <trk><name>Example gpx</name><number>1</number><trkseg>
    <trkpt lat="46.57608333" lon="8.89241667"><ele>2376</ele><time>2007-10-14T10:09:57Z</time></trkpt>
    <trkpt lat="46.57619444" lon="8.89252778"><ele>2375</ele><time>2007-10-14T10:10:52Z</time></trkpt>
    <trkpt lat="46.57641667" lon="8.89266667"><ele>2372</ele><time>2007-10-14T10:12:39Z</time></trkpt>
    <trkpt lat="46.57650000" lon="8.89280556"><ele>2373</ele><time>2007-10-14T10:13:12Z</time></trkpt>
    <trkpt lat="46.57638889" lon="8.89302778"><ele>2374</ele><time>2007-10-14T10:13:20Z</time></trkpt>
    <trkpt lat="46.57652778" lon="8.89322222"><ele>2375</ele><time>2007-10-14T10:13:48Z</time></trkpt>
    <trkpt lat="46.57661111" lon="8.89344444"><ele>2376</ele><time>2007-10-14T10:14:08Z</time></trkpt>
  </trkseg></trk>
</gpx>
```

</gpx>

Como se puede ver en el cuadro de texto anterior, el fichero GPX, en este caso, está formado por un total de siete puntos. Además de la latitud y la longitud, dichos puntos pueden contener otra información de interés como fechas, nombres, descripción... etc.

Capítulo 3. Planificación del proyecto y resumen de presupuesto

3.1 Planificación

En este capítulo se explicará todo lo relativo a la planificación del proyecto. Dicha planificación ha tenido más de una versión debido a diferentes motivos que serán explicados con cada una de las planificaciones.

3.1.1 Planificación inicial

En una primera reunión con el director del trabajo se estimó la carga total de horas en **700**, repartidas entre las diferentes fases que seguiría el proyecto. El desglose de porcentaje de horas estimado fue el siguiente, para la primera planificación:

Fase	Porcentaje de horas del total
Estudios iniciales	5%
Formación	10%
Análisis	15%
Diseño	5%
Implementación	35%
Pruebas	15%
Documentación	15%

Tabla 2 Desglose de porcentajes de horas en planificación inicial

3.1.2 Planificación inicial del proyecto

Teniendo la estimación del porcentaje de horas a realizar por fase, se realizó una primera planificación del proyecto, la cual se ilustra a continuación mediante el diagrama de Gantt y figuras extraídas del Project con algunas de las actividades más significativas.

▲ Proyecto Fin de Grado	700 horas	mar 01/11/16	dom 28/05/17
▲ Estudios iniciales	35 horas	mar 01/11/16	mar 08/11/16
Desarrollar propuesta de trabajo	10 horas	mar 01/11/16	jue 03/11/16
Evaluar alternativas	12 horas	jue 03/11/16	sáb 05/11/16
Resumen de aplicación	4 horas	sáb 05/11/16	dom 06/11/16
Justificación de alternativa	5 horas	dom 06/11/16	lun 07/11/16
Reuniones con el tutor del trabajo	4 horas	lun 07/11/16	mar 08/11/16
▲ Formación	70 horas	mar 08/11/16	jue 24/11/16
Curso Angular JS Coursera	70 horas	mar 08/11/16	jue 24/11/16

Figura 9 Estudios iniciales y formación de la planificación inicial

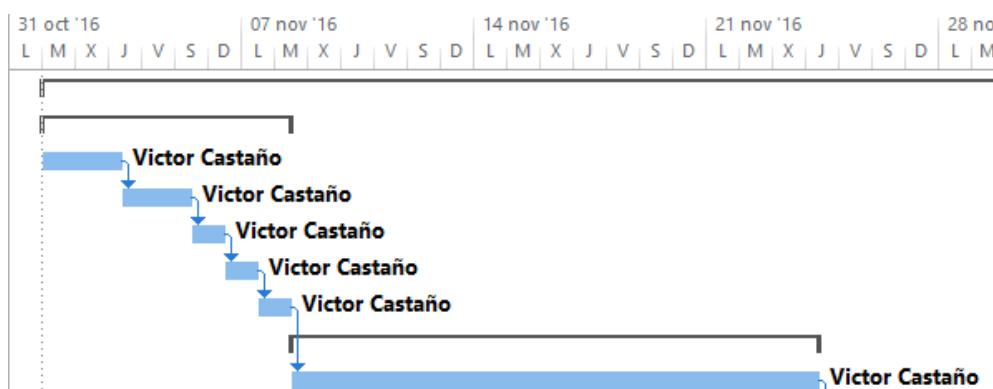


Figura 10 Gantt fases de estudios iniciales y formación

▲ Análisis	105 horas	jue 24/11/16	jue 29/12/16
▲ Requisitos funcionales y no funcionales	16 horas	jue 24/11/16	dom 27/11/16
Requisitos subsistema aplicación web	9 horas	jue 24/11/16	sáb 26/11/16
Requisitos subsistema aplicación móvil	7 horas	sáb 26/11/16	dom 27/11/16
▲ Estudio de interfaces de usuario	34 horas	dom 27/11/16	lun 05/12/16
Mapa de navegación web	2 horas	dom 27/11/16	lun 28/11/16
Mapa de navegación móvil	2 horas	lun 28/11/16	lun 28/11/16
Prototipos de pantalla aplicación web	15 horas	lun 28/11/16	vie 02/12/16
Prototipos de pantalla aplicación móvil	15 horas	vie 02/12/16	lun 05/12/16
Casos de uso nivel detallado	25 horas	lun 05/12/16	sáb 10/12/16
Casos de prueba	30 horas	dom 11/12/16	jue 29/12/16
▲ Diseño	35 horas	jue 29/12/16	dom 22/01/17
Diseño de la base de datos	35 horas	jue 29/12/16	dom 22/01/17

Figura 11 Fase de análisis y diseño planificación inicial

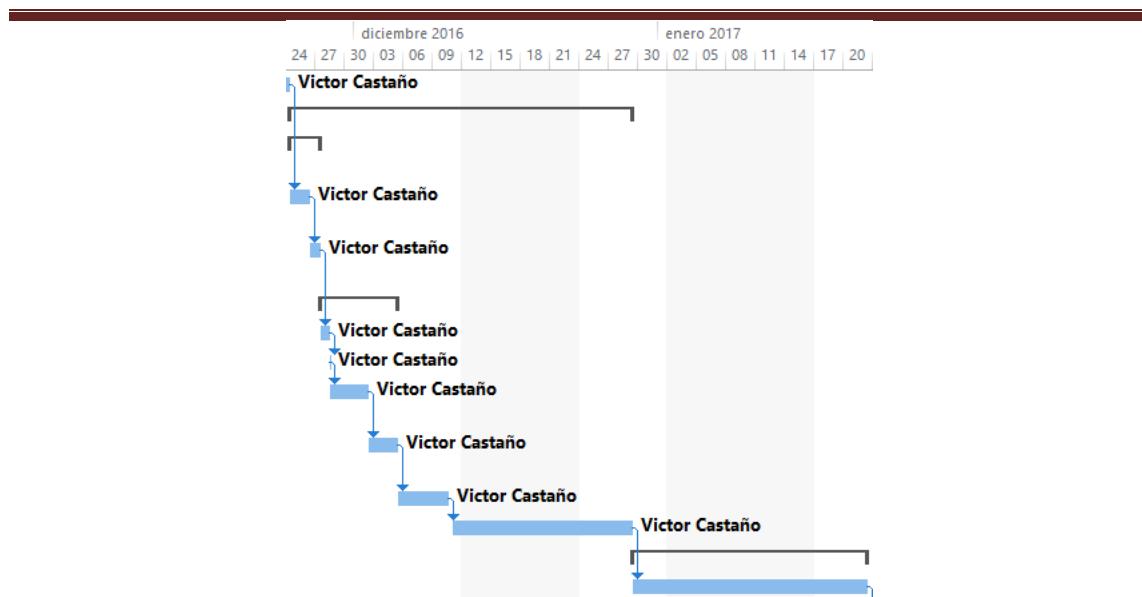


Figura 12 Gantt fase de análisis y diseño planificación inicial

Implementación	210 horas	dom 22/01/17	vie 10/03/17
Servidor	95 horas	dom 22/01/17	dom 12/02/17
Pruebas Junit	25 horas	dom 22/01/17	vie 27/01/17
Autenticación JWT	15 horas	sáb 28/01/17	mar 31/01/17
Modelo y persistencia	20 horas	mar 31/01/17	sáb 04/02/17
Servicios de negocio	25 horas	sáb 04/02/17	vie 10/02/17
Interfaces REST	10 horas	vie 10/02/17	dom 12/02/17
Aplicación web	40 horas	dom 12/02/17	mar 21/02/17
Diseño de routing	5 horas	dom 12/02/17	lun 13/02/17
Autenticación y servicios de usuarios y actividades	8 horas	lun 13/02/17	mié 15/02/17
Desarrollo de vistas	12 horas	mié 15/02/17	sáb 18/02/17
Pruebas Jasmine	15 horas	sáb 18/02/17	mar 21/02/17
Aplicación móvil	75 horas	mar 21/02/17	vie 10/03/17
Login y autenticación de usuario	15 horas	mar 21/02/17	vie 24/02/17
Vistas y activities	15 horas	vie 24/02/17	mar 28/02/17
Formación GreenDAO	10 horas	mar 28/02/17	jue 02/03/17
Modelo y anotaciones GreenDAO	8 horas	jue 02/03/17	sáb 04/03/17
Sistema de monitorización	12 horas	sáb 04/03/17	lun 06/03/17
Sistema de acciones offile	15 horas	lun 06/03/17	vie 10/03/17

Figura 13 Fase de implementación planificación inicial

Pruebas	105 horas	vie 10/03/17	dom 02/04/17
Ejecución de casos de prueba y valoración de resultados	105 horas	vie 10/03/17	dom 02/04/17
Documentación	140 horas	dom 02/04/17	dom 28/05/17

Figura 14 fase de pruebas y documentación planificación inicial

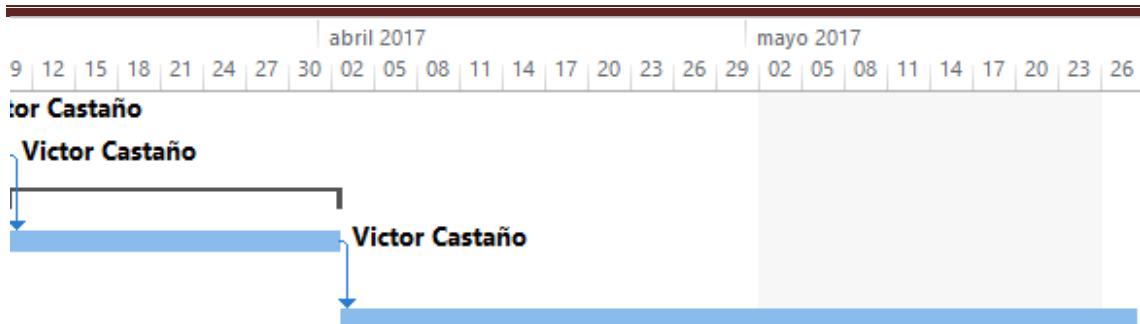


Figura 15 Gantt fases pruebas y documentación planificación inicial

Para tener una perspectiva global de la planificación inicial se adjunta en capturas el diagrama de Gantt completo.

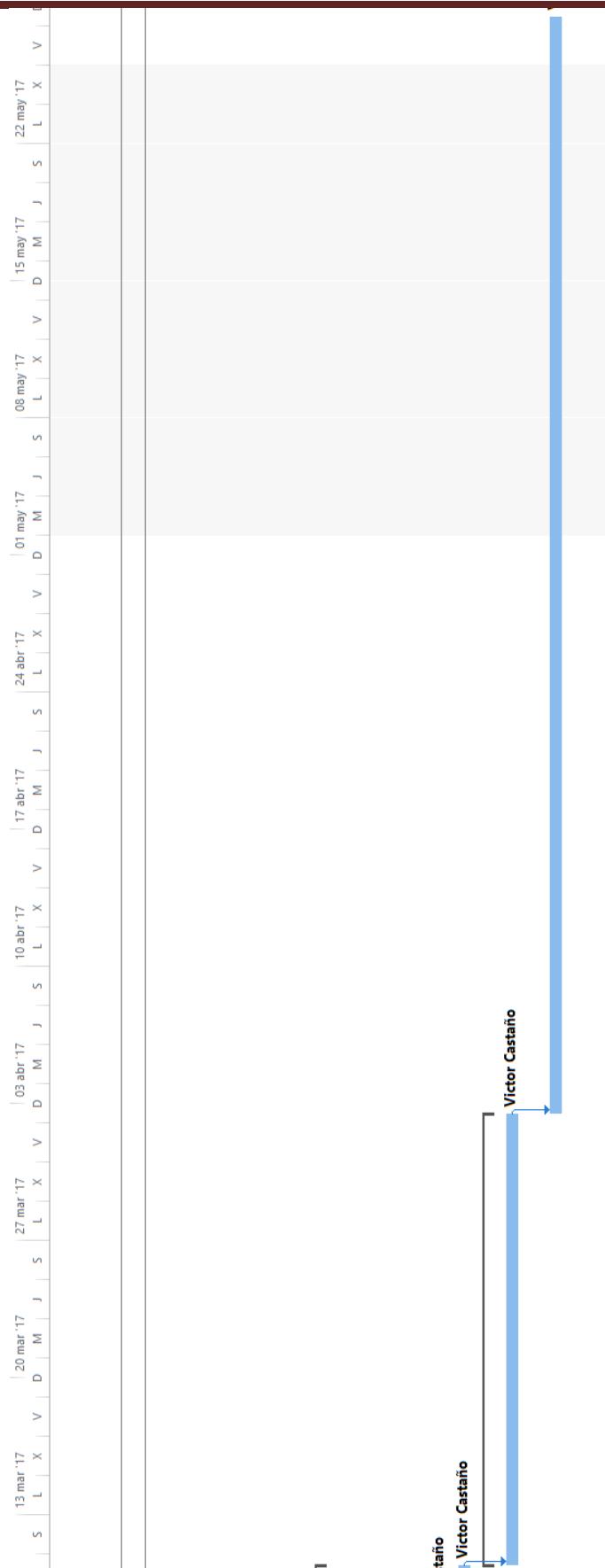


Figura 16 Gantt marzo - mayo planificación inicial

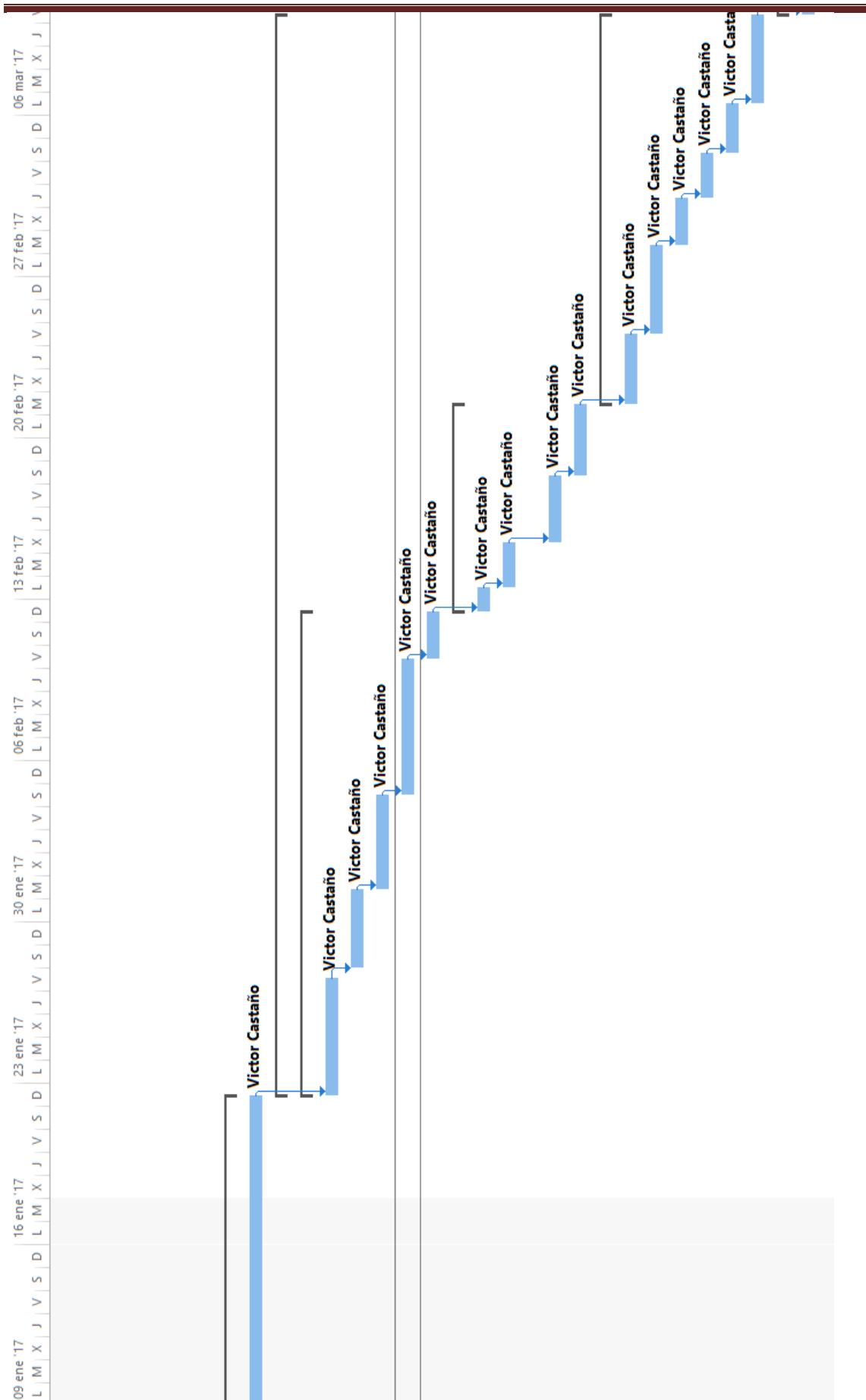


Figura 17 Gantt enero- marzo planificación inicial

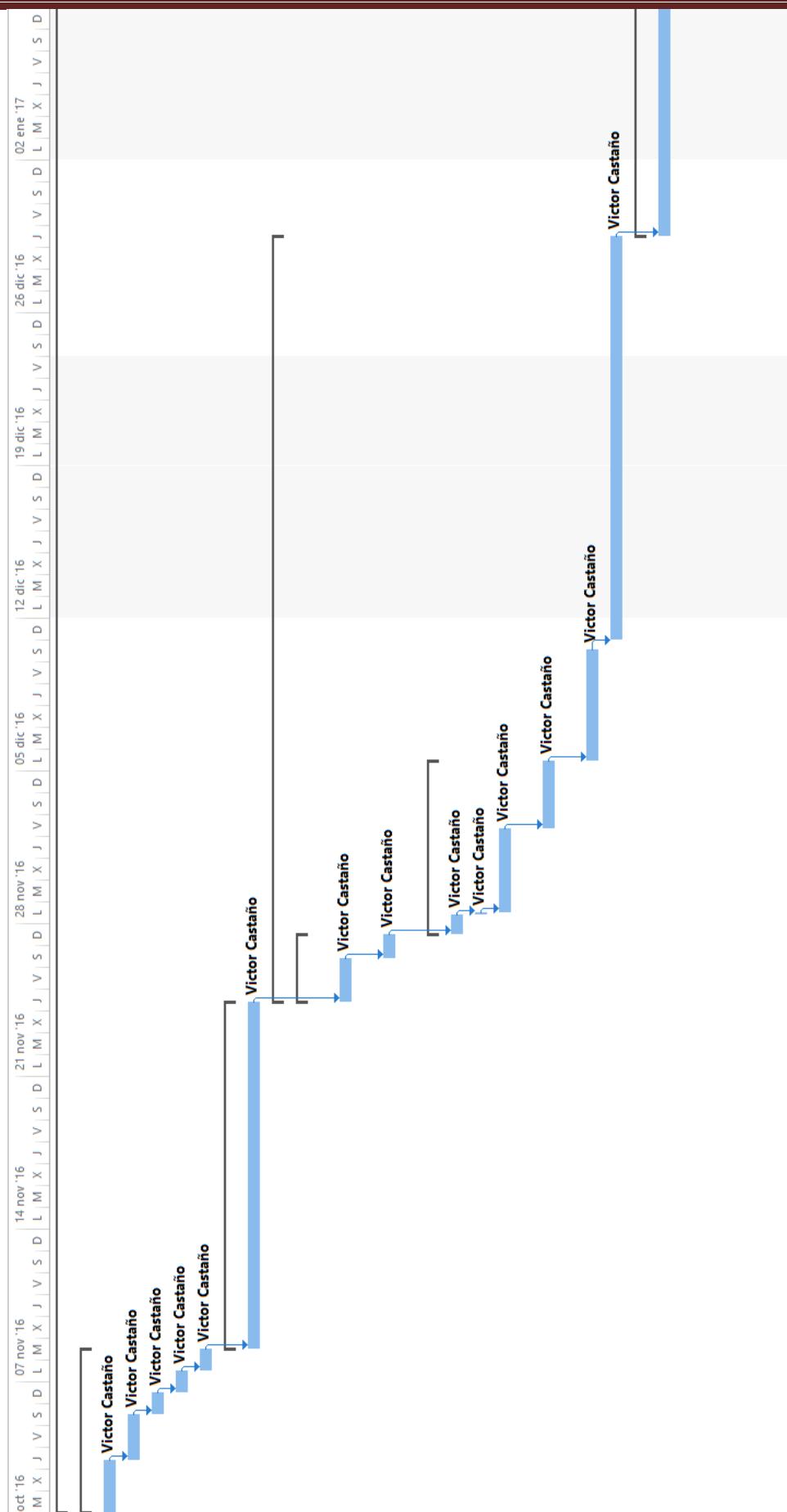


Figura 18 Gantt octubre-enero planificación inicial

3.1.3 Segunda planificación

Debido a diferentes problemas ocurridos durante los primeros meses de desarrollo del proyecto, me he visto abocado a la realización de una segunda planificación, al ver imposibilitada la opción de cumplir la primera e inicial.

La planificación inicial estaba pensada para finalizar el proyecto, aproximadamente, (con días de margen) para el mes de junio. Al sufrir retrasos en las primeras fases del proyecto, la segunda, y final, retrasó el fin del proyecto al mes de julio.

Los problemas que surgieron durante las primeras fases y conllevaron un retraso en la planificación fueron:

- Autenticación del servidor: el servidor para transferir datos con los clientes utiliza servicios web RESTFul. Estos servicios web, además de utilizar el protocolo HTTPS en las comunicaciones me vi obligado a llevar a cabo una autenticación. Dicha autenticación requirió de una formación extra que se tradujo en horas y días de retraso.
- Periodo de exámenes de diciembre-enero y entrega de trabajos en diferentes asignaturas: en el diseño de la planificación inicial no se había tenido en cuenta que el periodo de exámenes fuese a ser tan intrusivo, y finalmente ocasionó graves retrasos en el desarrollo del proyecto. Además de ello se decidió aportar una mayor cantidad de horas con semanas de excepción en la segunda planificación para el mes de mayo (coincidiendo con el segundo periodo de exámenes).

Por lo tanto, para la realización de la segunda planificación se partió de la inicial explicada anteriormente y se le añadieron semanas de excepción en enero y mayo además de horas previstas para formación en autenticación JWT.

3.1.4 Diario de trabajo

Durante la realización del proyecto se ha llevado actualizado un diario con las tareas en las que se invierte tiempo y la cantidad de tiempo invertido en cada una de ellas. Dicho diario, de manera detallada puede encontrarse en Diario de trabajo.

Teniendo la información precisa de las horas invertidas en cada fase se puede concluir mostrando un resumen del total de las horas invertidas en cada una de ellas. Las horas invertidas en reuniones con el tutor del proyecto van incluidas en la fase en la que se estuviese trabajando en el momento de la reunión.

En líneas generales ha habido una reunión con el tutor del proyecto cada vez que se ha cerrado una fase, con alguna añadida según se acercaban las fechas de finalización del mismo.

Fase	Horas invertidas	Porcentaje del total
Estudios iniciales	24	4%
Formación	64	11%
Análisis	77.25	13%

Diseño	17.5	3%
Implementación	241.5	40%
Pruebas	82	13%
Documentación	96.5	16%
TOTAL DE HORAS	602.75	100%

Tabla 3 Resumen de horas

A continuación, se muestra una representación gráfica de las horas invertidas en cada una de las fases del proyecto.

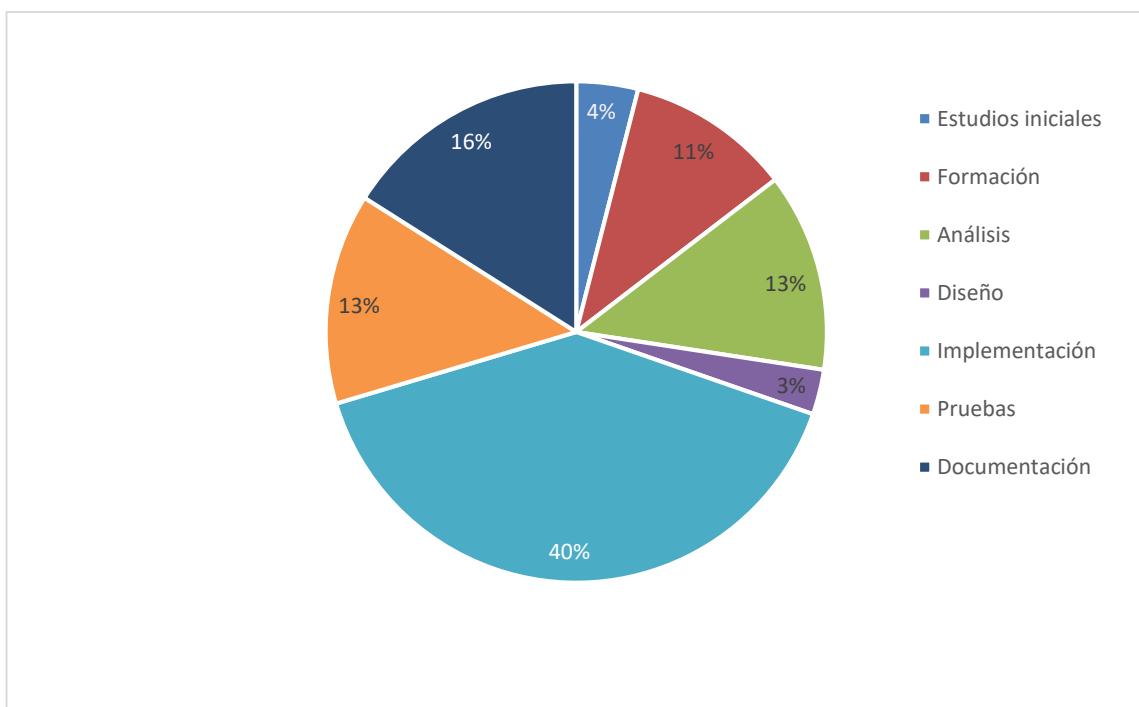


Figura 19 Gráfico circular comparación de horas por fases del proyecto

3.2 Resumen del Presupuesto

En este apartado del capítulo de la planificación se elaborará el presupuesto de ejecución del proyecto. Este presupuesto se establecerá en dos formas, el interno y el de cliente.

3.2.1 Coste de recursos de trabajo

A cada recurso de trabajo empleado para la ejecución del proyecto se le ha asignado un coste neto, y posteriormente bruto por hora.

Código	Unidad	Descripción	Precio
RT01	h	Analista	33,00€
RT02	h	Experto en pruebas	24,75€
RT03	h	Programador	15,75€

Tabla 4 Recursos de trabajo - presupuesto

3.2.2 Recursos materiales

En este apartado de incluyen todos los recursos materiales utilizados en el proyecto, incluyendo además las licencias necesarias. Debido a que el proyecto tiene una duración en horas específica, para los recursos materiales se ha calculado el factor de amortización, el cual aplicándolo obtendremos el precio a cobrar por cada uno de ellos.

$$\text{Factor de amortización} = \frac{\text{Duración del proyecto}}{\text{Días de amortización}}$$

Para tanto el hardware como el software suponemos una amortización de cuatro años. Esto se traduce en 1460 días, por lo que el factor de amortización, teniendo que la duración del proyecto es de 205 días, es de **0,1404**.

Código	Unidades	Descripción	Precio/ud	Amortización	Total
RM01	1	Ordenador estándar	500,00€	70,2€	70,2€
RM02	1	Licencia Microsoft Office 2016	100€	14,04€	14,04€
RM03	1	Libreta	1,50€	1,5€	1,5€
RM04	1	Bolígrafo	1,00€	1,0€	1,0€

Tabla 5 Recursos materiales - presupuesto

Derivado de estos recursos materiales obtenemos los costes en recursos de energía. Para el cálculo se ha tomado como referencia un consumo básico de kWh/hora de 0,12€. Esto se aplica al consumo de energía en los ordenadores y en los monitores, al que luego se le aplicará la cantidad de horas de uso.

Código	Unidad	Descripción	Precio/ud
RM05	h	Ordenador estándar + monitor	0,15€

Tabla 6 Coste de energía - presupuesto

3.2.3 Costes por fases

El coste hora por fase viene dado por la aplicación del coste bruto del recurso de trabajo asignado a esa fase más el coste por gastos de electricidad.

Fase	Horas	Coste/hora	Coste total
Análisis	77,25	33,15€	2.560,83€
Diseño	17,5	33,15€	580,12€
Implementación	241,5	15,90€	3.839,85€
Pruebas	82	24,90€	2.041,8€
Documentación	96,5	15,90€	1.534,35€

Tabla 7 Costes por fases presupuesto

3.2.4 Presupuesto interno

Medición	Concepto	Coste unitario	Total
1	Análisis y diseño	1	2.560,83€
1	Implementación	1	580,12€
1	Pruebas	1	3.839,85€
1	Documentación	1	2.041,8€
Presupuesto de ejecución material			86,74€
Beneficio 16%			1.457,49
Presupuesto de ejecución por administración			10.566,83€

Tabla 8 Presupuesto interno

3.2.5 Presupuesto de cliente

Para la realización del presupuesto de cliente se ha partido del presupuesto interno. Se ha decidido ocultar la fase de documentación, y, al igual que con los beneficios y el presupuesto de ejecución material, prorratearlo con las otras tres fases.

Medición	Concepto	Coste unitario	Total
1	Análisis y diseño	1	3.756,17€
1	Implementación	1	1.775,46€
1	Pruebas	1	5.035,19€
Presupuesto antes de impuestos			10.566,83
IVA (21%)			2.219,03
Presupuesto total estimado			12.785,86

Tabla 9 Presupuesto de cliente

Finalmente, como se observa en la tabla anterior, el presupuesto antes de impuestos coincide con el presupuesto de ejecución por administración. Aplicado el 21% de IVA queda un total de **12.785,86€**.

Capítulo 4. Análisis

En este capítulo se describirá todo el análisis de la aplicación realizado previamente a la implementación de la misma.

4.1 Definición del Sistema

4.1.1 Determinación del Alcance del Sistema

OCAM es una aplicación orientada a incrementar la seguridad en actividades al aire libre con múltiples participantes, por medio de la extracción constante de sus coordenadas GPS a través de una aplicación móvil y la monitorización en tiempo real a través de la aplicación web y de la misma aplicación móvil.

Los principales objetivos de la aplicación son:

- Registro y login de usuarios en la aplicación.
- Creación y gestión durante un ciclo de vida con diferentes estados de actividades.
- Inclusión de usuarios en actividades bajo roles de guía o excursionista.
- Extracción de información de posición de excursionistas en actividades que estén en curso.
- Monitorización de toda la información de participantes en tiempo real.
- Histórico y resúmenes de actividades realizadas.

4.1.2 Descripción de los subsistemas

La aplicación está compuesta por dos clientes y un servidor que se comunican a través de servicios web. Esto crea tres subsistemas que se describen a continuación.

Cliente **aplicación web** que consume ciertos datos del servidor a través de interfaces REST y que alberga parte de la funcionalidad deseada para la aplicación.

Cliente **aplicación móvil** nativa para SO Android que consume ciertos datos del servidor a través de interfaces REST. Se complementa con el cliente web para conformar el total de la funcionalidad de la aplicación.

El **servidor** es el encargado de gestionar toda la información que envían y consumen a través de las interfaces REST tanto el cliente móvil como desde la web. Recibe la información, la trata, la persiste y la retorna a cada uno de los clientes de la manera que sea solicitada.

En la siguiente figura se puede apreciar el esquema de funcionamiento a muy alto nivel de los subsistemas, y cuál es la dirección de las comunicaciones.

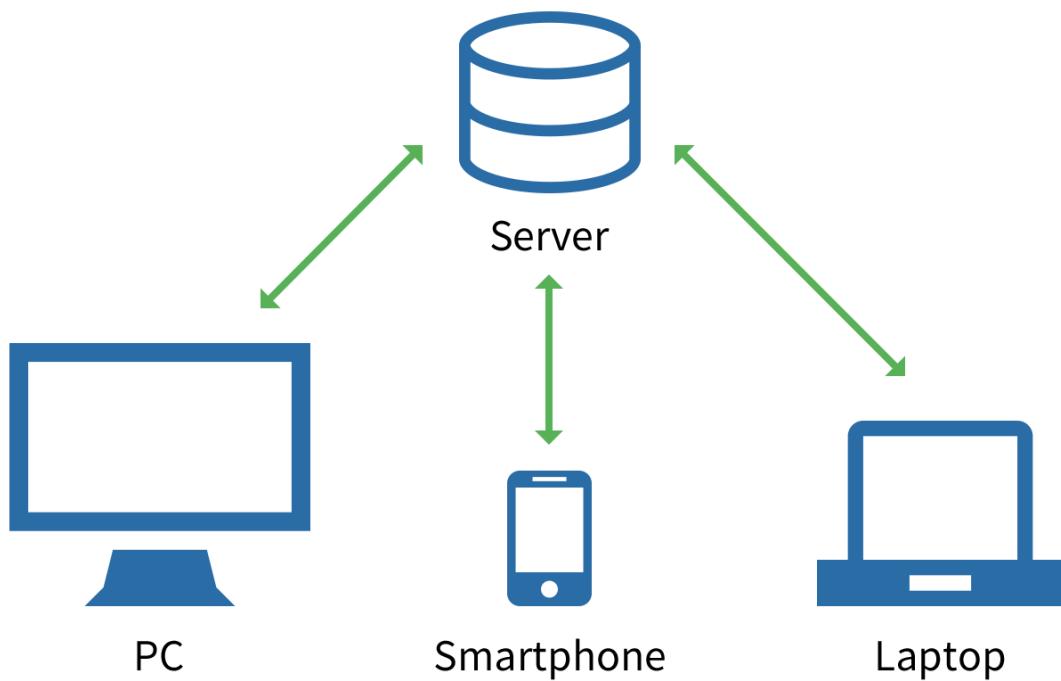


Figura 20 Esquema de funcionamiento subsistemas

4.2 Requisitos del Sistema

En este apartado del análisis del sistema se expondrán los requisitos funcionales y no funcionales. Se identificarán tanto los requisitos de la aplicación web como los requisitos de la aplicación móvil.

Cada requisito estará identificado por un identificador único y la descripción del propio requisito. Así mismo, los identificadores del subsistema web y móvil no son el mismo.

4.2.1 Obtención de los Requisitos del Sistema

4.2.1.1 *Requisitos funcionales subsistema web*

RFW1. La aplicación permitirá a un usuario no identificado registrarse aportando los siguientes datos obligatorios:

RFW1.1. Correo electrónico

RFW1.1.1. La aplicación verificará que no existe ningún usuario registrado con el mismo correo electrónico

RFW1.1.2. La aplicación verificará que el correo contenga una '@' y un dominio.

RFW1.2. Nombre de usuario

RFW1.2.1. La aplicación verificará que no exista ningún usuario registrado en la aplicación con el mismo login.

RFW1.3. Password

RFW1.4. Repetir password

RFW1.4.1. La aplicación verificará que ambas passwords coinciden

RFW1.5. Si alguna de las validaciones anteriores fallase la aplicación notificará al usuario con aquel campo que necesite ser cambiado.

RFW1.6. En caso de validar todos los campos la aplicación registrará al usuario con los datos aportados.

RFW1.6.1. La aplicación enviará un correo electrónico a la dirección aportada en el registro con un enlace para validar la cuenta.

RFW1.6.2. La aplicación cambiará el estado de la cuenta a validada automáticamente al entrar en el enlace proporcionado.

RFW2. La aplicación permitirá iniciar sesión a un usuario no identificado por medio de los siguientes campos obligatorios:

RFW2.1. Login

RFW2.2. Contraseña

RFW2.3. La aplicación comprobará que existe un usuario registrado y validado con esas credenciales.

RFW2.3.1. En caso de no ser correctas se notificará del fallo de inicio de sesión

RFW2.3.2. Si las credenciales son correctas iniciará en sesión al usuario y lo redirecciónará a la lista de actividades pendientes.

RFW3. La aplicación permitirá ver el listado de actividades pendientes de realizar a cualquier usuario

RFW3.1. Las actividades pendientes de realizar son las que se encuentran en estado PENDING o RUNNING y no están eliminadas

RFW3.2. De cada actividad mostrará en el listado:

RFW3.2.1. Fecha prevista de realización

RFW3.2.2. Plazas máximas

RFW3.2.3. Descripción corta de la actividad

RFW3.2.4. Estado de la actividad

RFW3.3. A través del listado la aplicación permitirá acceder al detalle de cada actividad

RFW3.3.1. Para poder acceder al detalle el usuario tendrá que estar identificado en la aplicación

RFW3.3.2. En caso de no estar identificado la aplicación notificará al usuario que debe iniciar sesión antes de ver el detalle de una actividad

RFW4. La aplicación permitirá ver las actividades realizadas a cualquier usuario identificado en la aplicación

RFW4.1. Una actividad realizada es aquella en estado CLOSED

RFW4.2. Cada usuario solo verá las actividades para las cuales han enviado reportes.

RFW4.3. De cada actividad mostrará los datos que se describe en RFW3.2

RFW4.3.1. En el caso de la fecha mostrará la fecha de apertura de la actividad

RFW4.4. Para cada actividad se podrá acceder a la monitorización de la misma tal y como se muestra en RFW12.5 para ver el resumen de la misma

RFW4.4.1. En caso de ser guía de la actividad verá los datos de todos los participantes de la misma

RFW4.4.2. En caso de no ser guía de la actividad solo podrá ver el resumen de su participación en la misma

RFW5. La aplicación permitirá a un usuario identificado ver los datos de su cuenta

RFW5.1. Se mostrarán:

RFW5.1.1. Nombre de usuario

RFW5.1.2. Correo electrónico

RFW6. La aplicación permitirá recuperar la contraseña a un usuario no identificado

RFW6.1. La aplicación solicitará el correo electrónico de la cuenta

RFW6.1.1. Comprobará si el correo electrónico pertenece a una cuenta registrada y validada

RFW6.1.2. En caso de existir se enviará una nueva contraseña generada aleatoriamente al correo electrónico de la cuenta de usuario

RFW7. La aplicación permitirá cambiar la contraseña a un usuario identificado

RFW7.1. Se pedirán como datos obligatorios

RFW7.1.1. Contraseña actual

RFW7.1.2. Contraseña nueva

RFW7.1.3. Repetición de contraseña nueva

RFW7.2. La aplicación verificará que la contraseña actual es correcta

RFW7.3. La aplicación verificará que las dos contraseñas nuevas coinciden

RFW7.4. En caso de validarse lo anterior la aplicación actualizará la contraseña del usuario.

RFW8. La aplicación permitirá ver a un usuario identificado los detalles de una actividad

RFW8.1. La aplicación mostrará los siguientes datos:

RFW8.1.1. Descripción corta

RFW8.1.2. Organizador de la actividad

RFW8.1.2.1. Mostrará el correo electrónico del usuario que creó la actividad

RFW8.1.3. Descripción larga de la actividad

RFW8.1.4. Enlace MIDE

RFW8.1.5. Estado de la actividad

RFW8.1.6. Plazas máximas

RFW8.1.7. Fecha de comienzo prevista para la actividad

RFW8.1.8. Guías de la actividad

RFW8.1.8.1. De cada guía mostrará el nombre de usuario del mismo

RFW8.1.9. Track de la ruta

RFW8.1.9.1. La aplicación mostrará el track sobre un mapa mediante una polylinea

RFW8.1.9.2. La aplicación permitirá cambiar el tipo de mapa entre: Raster, PNOA, OSM, Raster FR, Satélite y Mapa

- RFW8.1.9.3.** La aplicación mostrará mediante un marcador el punto de inicio y el punto de fin de la ruta.
- RFW8.1.9.4.** La aplicación mostrará sobre la línea de la ruta los puntos que la forman
- RFW8.1.9.4.1.** La aplicación mostrará los puntos cuando se alcance un zoom de X
- RFW8.1.9.4.1.1.** X es numérico e inicialmente vale 17
- RFW8.1.9.4.2.** La aplicación ocultará los puntos al alejarse del nivel de zoom X
- RFW8.1.9.5.** La aplicación mostrará en todo momento las coordenadas sobre las que se encuentra el cursor sobre el mapa
- RFW8.1.9.6.** La aplicación mostrará una rejilla sobre el mapa formada por cuadrados de ZxZ pixels de lado cada uno
- RFW8.1.9.6.1.** Z es numérico e inicialmente vale 256
- RFW8.1.9.7.** La aplicación mostrará las coordenadas en cada intersección de la rejilla
- RFW8.1.9.8.** La aplicación mostrará en todo momento la distancia en metros entre vértices de la rejilla.
- RFW9.** La aplicación permitirá crear una nueva actividad
- RFW9.1.** Para crear una actividad se necesitará estar identificado en la aplicación
- RFW9.2.** La aplicación pedirá los siguientes datos obligatorios:
- RFW9.2.1.** Descripción breve
- RFW9.2.2.** Fecha de comienzo
- RFW9.2.2.1.** Solo se permitirán fechas de comienzo superiores a la fecha actual
- RFW9.2.3.** Track de la ruta
- RFW9.2.3.1.** Solo se permitirán ficheros con extensión GPX
- RFW9.2.4.** Guías de la actividad
- RFW9.2.4.1.** La aplicación permitirá añadir guías sin límite identificándolos a través del nombre de usuario
- RFW9.2.4.1.1.** La aplicación comprobará que el nombre de usuario introducido corresponde con un usuario registrado en la aplicación
- RFW9.2.4.2.** La aplicación permitirá eliminar guías añadidos previamente
- RFW9.2.4.3.** La aplicación añadirá por defecto como guía al usuario que crea la actividad
- RFW9.2.4.4.** Mínimo se deberá añadir un guía a la actividad
- RFW9.3.** La aplicación pedirá los siguientes datos opcionales:
- RFW9.3.1.** Descripción de la ruta
- RFW9.3.2.** Enlace MIDE
- RFW9.3.3.** Plazas máximas
- RFW9.4.** La aplicación verificará que o bien el enlace MIDE o bien la descripción de la ruta, o ambos, han sido introducidos
- RFW9.5.** La aplicación verificará que todos los campos son correctos y en caso de no serlos notificará al usuario indicando cual debe ser corregido
- RFW9.5.1.** En caso de ser correctos la aplicación creará la actividad en estado PENDING
- RFW10.** La aplicación permitirá editar una actividad
- RFW10.1.** La actividad deberá estar en estado PENDING o RUNNING
- RFW10.2.** Solo podrá editar una actividad el usuario que la creó.
- RFW10.3.** Todos los campos de la actividad serán editables
- RFW10.4.** La aplicación permitirá editar los campos tal y como se describe en RFW9.2
- RFW11.** La aplicación permitirá eliminar una actividad en estado PENDING
- RFW11.1.** Solo podrá eliminar una actividad el usuario que la creó
- RFW11.2.** La aplicación realizará un borrado lógico de la actividad
- RFW12.** La aplicación permitirá a un usuario identificado monitorizar una actividad

- RFW12.1.** La actividad deberá estar en estado RUNNING
- RFW12.2.** Para acceder a la monitorización la aplicación solicitará al usuario la contraseña de la actividad
- RFW12.2.1.** En caso de no coincidir con la contraseña de la actividad se notificará al usuario del fallo
- RFW12.3.** La aplicación mostrará el track de la ruta tal y como se describe en RFW8.1.9
- RFW12.4.** La aplicación mostrará para cada participante los siguientes datos:
 - RFW12.4.1.** Correo electrónico
 - RFW12.4.2.** Hora del último reporte enviado
- RFW12.5.** La aplicación permitirá mostrar sobre el mapa para cada participante la trayectoria que ha seguido
 - RFW12.5.1.** La trayectoria se mostrará tal y como se describe en RFW8.1.9
 - RFW12.5.2.** Al mostrar la trayectoria el mapa se moverá a la posición de la misma
- RFW12.6.** La aplicación permitirá mostrar sobre el mapa para cada participante un marcador con la última posición conocida.
 - RFW12.6.1.** El mapa se moverá automáticamente a la posición del marcador con la posición del excursionista.
- RFW12.7.** La aplicación permitirá recargar la información de las posiciones de cada uno de los participantes de la actividad.

RFW13. La aplicación permitirá cerrar sesión a un usuario identificado.

RFW14. La aplicación permitirá a un usuario identificado eliminar su cuenta

RFW14.1. La aplicación solicitará al usuario la confirmación de la eliminación

RFW14.2. En caso de confirmar la eliminación realizará un borrado lógico de la cuenta

4.2.1.2 Requisitos no funcionales subsistema web

RNFW1. Será necesaria conexión a internet para acceder a la aplicación.

RNFW2. El sistema garantizará que la conexión y los datos que se envían al servidor se hacen de forma segura.

RNFW3. La aplicación será usable, no requerirá de experiencia previa para la utilización de todas las funcionalidades de la misma.

RNFW4. La aplicación podrá usarse en los navegadores webs más usados actualmente.

RNFW5. La aplicación será responsive, se adaptará a diferentes resoluciones de pantalla incluyendo dispositivos móviles.

RNFW6. El sistema garantizará que un usuario no podrá acceder a datos de otros usuarios de la aplicación.

RNFW7. La aplicación estará internacionalizada en al menos dos idiomas: español e inglés.

4.2.1.3 Requisitos funcionales subsistema aplicación móvil

RFM1. La aplicación permitirá a un usuario no identificado registrarse

RFM1.1. Solicitará los siguientes campos obligatorios:

RFM1.1.1. Nombre de usuario (login)

RFM1.1.2. Correo electrónico

RFM1.1.2.1. Deberá tener una '@' y un dominio

RFM1.1.3. Password

RFM1.1.4. Repite password

RFM1.2. Verificará que las dos passwords coinciden entre ellas

RFM1.3. Verificará que tanto el nombre de usuario como el correo electrónico no están en uso por ningún otro usuario registrado

RFM1.4. En caso de validarse los campos introducidos registrará al usuario y enviará un correo electrónico con un enlace para validar la cuenta tal y como se describe en RFW1.6

RFM2. La aplicación permitirá hacer login en el sistema a un usuario no identificado y con conexión a internet

RFM2.1. Deberá aportar los siguientes campos obligatorios:

RFM2.1.1. Login

RFM2.1.2. Password

RFM2.2. Verificará que existe un usuario registrado y validado con las credenciales aportadas

RFM2.3. Permitirá al usuario guardar los datos de la cuenta (vincular la cuenta)

RFM2.3.1. En caso de tener los datos guardados la aplicación iniciará al usuario en sesión automáticamente sin introducir las credenciales

RFM3. La aplicación iniciará sesión a un usuario sin conexión a internet si tiene la cuenta vinculada

RFM3.1. Iniciará sesión al usuario automáticamente y le notificará del estado de la conectividad del dispositivo

RFM4. La aplicación mostrará el listado de actividades pendientes de realización a un usuario identificado

RFM4.1. Una actividad pendiente de realización es aquella que esté en estado PENDING o RUNNING.

RFM4.2. Permitirá filtrar las actividades por fecha de realización, quedándose con aquellas con fecha superior a la introducida por el usuario.

RFM4.3. Para cada actividad mostrará:

RFM4.3.1. Descripción

RFM4.3.2. Fecha de inicio prevista

RFM4.3.3. Estado de la actividad

RFM4.3.4. Detalle de la actividad en el que se mostrará accediendo al mismo:

RFM4.3.4.1. Descripción de la actividad

RFM4.3.4.2. Plazas máximas de la actividad

RFM4.3.4.3. Enlace al MIDE de la actividad

RFM4.3.4.3.1. A través del enlace la aplicación abrirá el navegador del dispositivo con el enlace.

RFM4.4. La aplicación permitirá recargar los datos de las actividades siempre y cuando el dispositivo tenga conexión a internet.

RFM4.4.1. Cada vez que se recargue la lista de actividades se guardarán los datos de manera local en el dispositivo.

RFM5. La aplicación permitirá acceder a la lista de actividades sin conexión mostrándose las últimas almacenadas.

RFM6. La aplicación permitirá acceder al detalle de una actividad

RFM6.1. En el detalle mostrará:

RFM6.1.1. Estado

RFM6.1.2. Email del creador de la actividad

RFM6.1.3. Descripción corta de la actividad

RFM6.1.4. Fecha prevista de la actividad

RFM6.1.5. Plazas máximas de la actividad

RFM6.1.6. Track de la ruta

RFM6.1.6.1. El track de la ruta se mostrará según lo descrito en RFW8.1.9

RFM6.2. En caso de ser guía de la actividad notificará al usuario de que lo es

RFM6.3. En caso de estar unido notificará al usuario de que está unido a la actividad

RFM6.4. En caso de ser guía de la actividad permitirá

RFM6.4.1. Iniciar la actividad si está en estado PENDING

RFM6.4.1.1. Solicitará al usuario una contraseña que deberá ser entre X e Y caracteres

RFM6.4.1.1.1. X e Y son numéricos e inicialmente valen 4 y 8 respectivamente

RFM6.4.1.2. En caso de validarse la contraseña introducida dará por iniciada la actividad cambiándola a estado RUNNING.

RFM6.4.1.3. Establecerá la fecha de inicio de la actividad a la fecha en la que se abre.

RFM6.4.2. Cerrar la actividad si está en estado RUNNING

RFM6.4.2.1. Cambiará la actividad a estado CLOSED

RFM7. En caso de no estar unido a la actividad y estar ésta en estado RUNNING permitirá unirse a la misma.

RFM7.1. Solicitará al usuario la password de acceso

RFM7.1.1. En caso de ser incorrecta notificará al usuario del error

RFM7.1.2. En caso de ser correcta incluirá al usuario en la actividad y le notificará de la acción.

RFM7.2. En caso de no haberse proporcionado ya a aplicación solicitará los permisos de localización del dispositivo

RFM7.3. La aplicación notificará al usuario con el uso que se le darán a la posición se que le extraerá al dispositivo durante la monitorización de la actividad.

RFM7.4. La aplicación comprobará el estado del GPS del dispositivo

RFM7.4.1. En caso de estar desactivado notificará al usuario de la necesidad de activarlo, previamente a unirse a la actividad

RFM8. La aplicación permitirá ver los datos de usuario a un usuario identificado

RFM8.1. Mostrará los siguientes datos:

RFM8.1.1. Email

RFM8.1.2. Nombre de usuario

RFM8.2. Permitirá cambiar la contraseña introduciendo de manera obligatoria para ello:

RFM8.2.1. Contraseña actual

RFM8.2.2. Contraseña nueva

RFM8.2.3. Repetición de contraseña nueva

RFM8.3. Verificará que la contraseña actual es correcta

RFM8.4. Verificará que las contraseñas nuevas coinciden

RFM8.5. En caso de validar los campos introducidos la aplicación actualizará la password del usuario

RFM9. La aplicación permitirá monitorizar una actividad a todo usuario unido a la misma

RFM9.1. La aplicación deberá en estado RUNNING

RFM9.2. La monitorización se mostrará tal y como se describe en RFW12

RFM9.3. La aplicación mostrará en la pantalla de monitorización el número de reportes enviados y el número de reportes encolados de manera local

RFM10. La aplicación permitirá cerrar sesión a un usuario identificado.

RFM11. La aplicación recogerá cada X minutos la posición del dispositivo

RFM11.1. X es un número entero e inicialmente vale 4

RFM11.2. X será configurable por el usuario

RFM11.2.1. La aplicación solamente permitirá valores entre Y y Z.

RFM11.2.2. Y y Z son valores numéricos e inicialmente valen 1 y 360 respectivamente

RFM11.3. Lo hará siempre y cuando el usuario esté unido a una actividad en estado RUNNING

RFM11.4. Previo a obtener la posición del dispositivo comprobará que los ajustes del mismo son adecuados para obtener la posición GPS.

RFM11.4.1. En caso de no ser ajustes adecuados no obtendrá la posición y notificará al usuario del error.

RFM11.5. De cada reporte recogerá

RFM11.5.1. La posición del dispositivo

RFM11.5.2. La fecha en la que se recogió el reporte

RFM11.6. En caso de no tener conectividad a internet la aplicación almacenará el reporte de manera local.

RFM11.6.1. La aplicación notificará al usuario del número de reportes encolados.

RFM11.7. Cada vez que se recoja un reporte la aplicación lo enviará al servidor

RFM11.7.1. En caso de tener reportes encolados de manera local los enviará junto al recogido en ese momento.

RFM11.7.2. Con cada reporte enviado al servidor la aplicación notificará al usuario de la hora del último reporte enviado

RFM12. La aplicación permitirá llevar a cabo acciones sin tener conexión

RFM12.1. Almacenará las acciones de manera local

RFM12.2. Procesará todas las acciones en el orden en que se realizaron una vez se reconecte el dispositivo a internet

RFM12.3. Las acciones que se podrán llevar a cabo sin conexión son:

RFM12.3.1. Unirse a una actividad

RFM12.3.2. Cerrar una actividad

RFM12.3.3. Abandonar una actividad

RFM12.3.4. Abrir una actividad

RFM13. La aplicación permitirá a un usuario unido a una actividad abandonarla

RFM13.1. La actividad tendrá que estar en estado RUNNING.

RFM13.2. El usuario dejaría de formar parte de la actividad

RFM13.3. Se interrumpiría la monitorización para el usuario

4.2.1.4 Requisitos no funcionales subsistema aplicación móvil

RNFM1. Las comunicaciones que se realicen con el servidor desde la aplicación móvil serán seguras.

RNFM2. La versión mínima de sistema operativo Android necesaria para ejecutar la aplicación es 5.0 Lollipop.

RNFM3. Para poder hacer uso de la funcionalidad completa de la aplicación el dispositivo móvil deberá contar con conexión GPS.

RNFM4. Es necesario conexión a internet para cierta funcionalidad de la aplicación.

RNFM5. La aplicación será usable, no requerirá de experiencia previa para la utilización de la misma.

RNFM6. La aplicación móvil no realizará consumos de batería altos durante períodos de funcionamiento siendo monitorizado el dispositivo.

4.3 Modelo de dominio

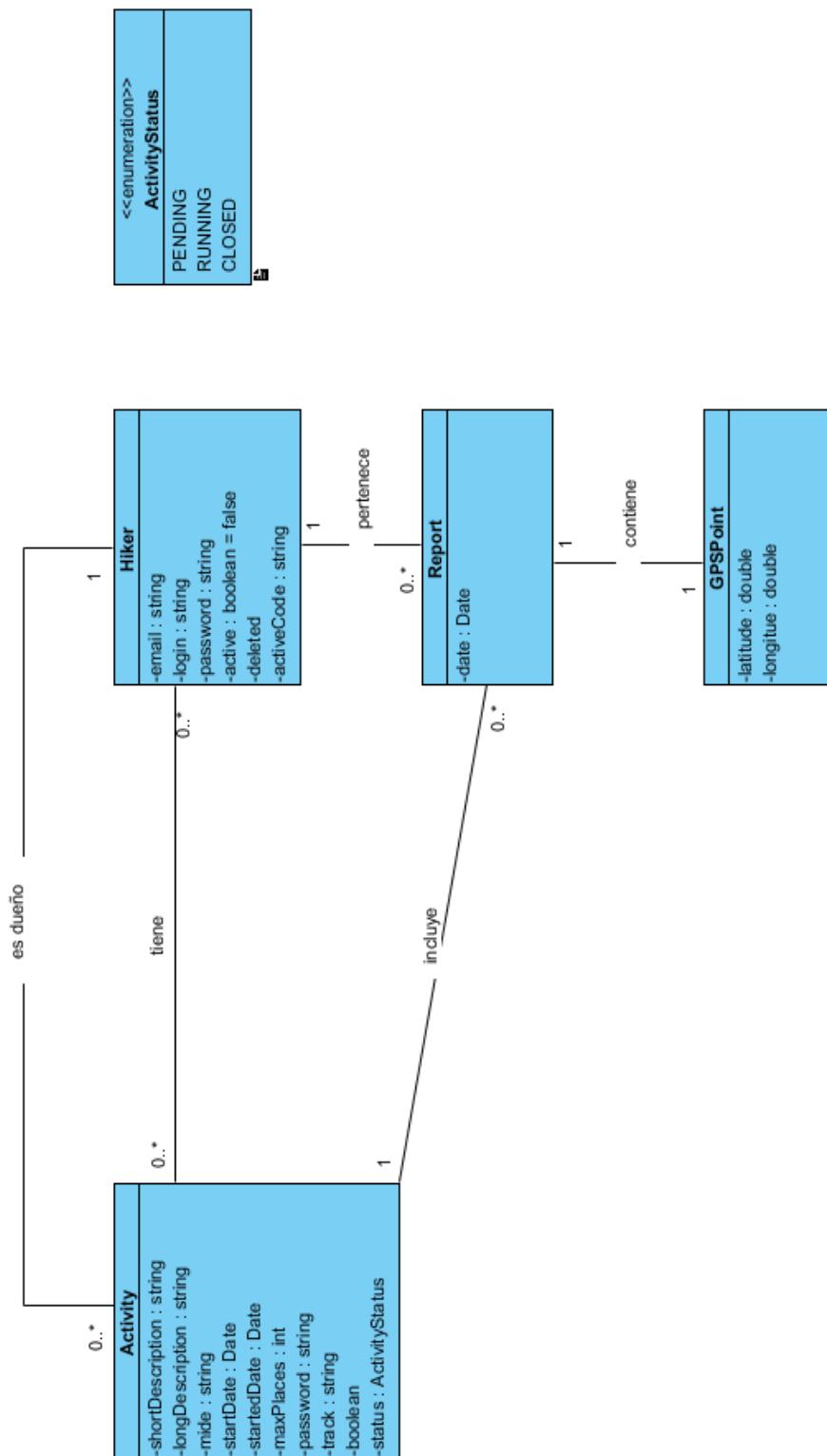


Figura 21 Modelo de dominio

- Activity: encapsula la información de cualquier actividad que se realice en la aplicación.
- Hiker: encapsula la información de los excursionistas, participantes de actividades
- Report: encapsula la información relativa a un reporte generado por un excursionista.
- GPSPoint: punto GPS que completa la información de un reporte
- ActivityStatus: diferentes estados por los que puede pasar una actividad

4.4 Análisis de Casos de Uso y escenarios

4.4.1 Identificación de actores del sistema

Tanto para el subsistema aplicación web como el subsistema aplicación móvil, se distinguen los siguientes actores:

Actor	Descripción
Usuario invitado	Usuario que interactúa con el sistema y que no está identificado / vinculado con la aplicación. Tendrá acceso a una funcionalidad limitada en cada uno de los subsistemas
Usuario identificado	Usuario previamente identificado en el sistema ya sea, en el caso del subsistema web, por medio del login, o en el caso del subsistema móvil por medio de la vinculación. Tendrá acceso a la funcionalidad total del sistema bajo permisos en determinados casos de uso.
Guía de la actividad	Se trata de una generalización del actor usuario identificado. En este caso el guía de una actividad es aquel que puede iniciarla y darla por finalizada.

Tabla 10 Identificación de actores del sistema

4.4.2 Especificación de casos de uso y escenarios subsistema aplicación web

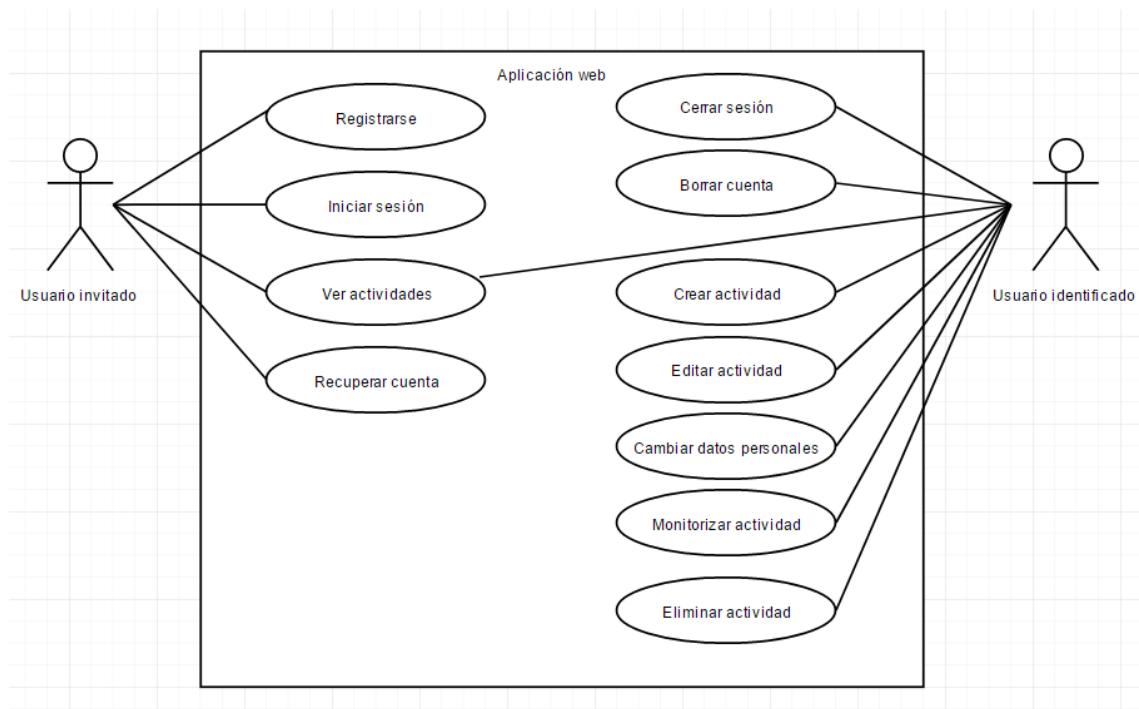


Figura 22 Diagrama casos de uso subsistema aplicación web

Caso de uso	Descripción
62	Víctor Castaño Gutiérrez Escuela de Ingeniería Informática - Universidad de Oviedo

Registrarse	Un usuario invitado podrá registrarse en la aplicación aportando para ello el correo electrónico, nombre apellidos y contraseña.
Ver actividades	Listado de las actividades en curso o pendientes de realización, así como los datos asociados a las mismas.
Iniciar sesión	A través de las credenciales, iniciar sesión en el sistema.
Recuperar cuenta	A través del correo electrónico solicitar una nueva contraseña para la cuenta
Cerrar sesión	Un usuario previamente identificado podrá cerrar sesión del sistema
Borrar cuenta	Realizar un borrado lógico de la cuenta de usuario
Crear actividad	Crear una nueva actividad en el sistema pendiente de realización aportando para ello los datos de la misma.
Editar actividad	Editar una actividad existente en el sistema. Modificación de cualquiera de sus datos asociados.
Cambiar datos personales	Modificación de los datos personales de un usuario registrado
Monitorizar actividad	Monitorización de una actividad, ver toda la información de las personas que son monitorizadas, así como su posición a tiempo real y el track de la ruta.
Eliminar actividad	Realizar un borrado lógico de la actividad

Tabla 11 Casos de uso subsistema aplicación web

4.4.2.1 CU-W-1 Registrarse

Caso de uso	Registrarse
Actores	Usuario invitado
Iniciado por	Usuario invitado
Tipo	Primario
Referencias	-
Precondiciones	El usuario no debe estar registrado en el sistema El usuario no debe estar identificado en el sistema
Postcondiciones	El nuevo usuario debe tener un correo electrónico asociado (único) y una contraseña.
Requisitos que satisface	RFW1
Propósito	Registrar a un usuario no identificado ni registrado previamente en el sistema.
Resumen de alto nivel	Un usuario accederá al sistema, introducirá un correo electrónico, contraseña, nombre y apellidos y creará su cuenta.
Descripción detallada	Flujo de acciones del camino básico
ACTOR	SISTEMA
1. El usuario invitado introduce su correo electrónico, su contraseña y de nuevo su contraseña.	2. Comprueba que el correo electrónico es válido (cumple con el formato requerido). Comprueba que el correo electrónico no está en uso previamente por otro usuario ya registrado en el sistema.

3. Envía los datos confirmando así su registro	4. Crea la cuenta en el sistema introduciendo sus datos en la base de datos. Redirecciona al usuario a la página principal de la aplicación
Flujo de acciones de caminos alternativos	
El sistema puede no validar los datos del cliente por correo electrónico inválido o en uso previamente. Contraseña y repetir contraseña no coinciden	

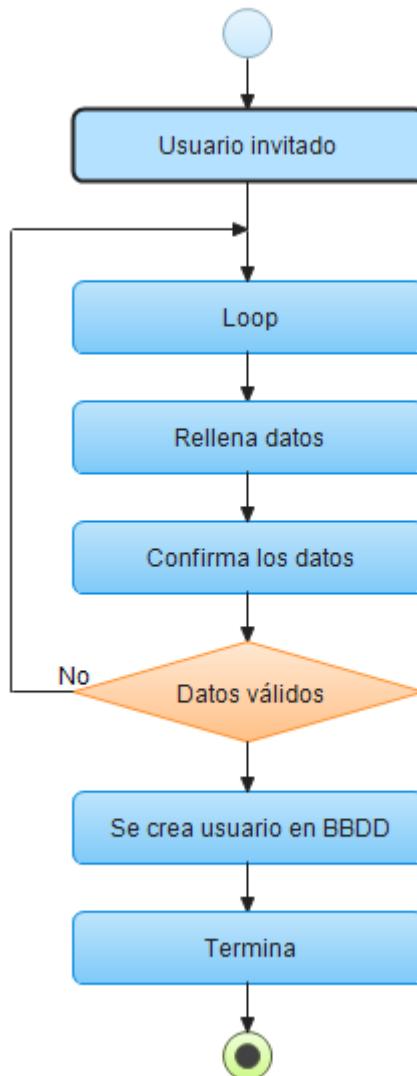


Figura 23 Diagrama de actividad - CU-W-1 Registrarse

4.4.2.2 CU-W-2 Iniciar sesión

Caso de uso	Iniciar sesión
Actores	Usuario invitado
Iniciado por	Usuario invitado
Tipo	Primario
Referencias	-
Precondiciones	El usuario deberá estar registrado en el sistema

	El usuario no deberá tener la sesión iniciada
Postcondiciones	El usuario estará identificado en el sistema
Requisitos que satisface	RFW2
Propósito	
Iniciar sesión en el sistema a través de las credenciales	
Resumen de alto nivel	
El usuario accede al sistema, introduce su correo y contraseña y inicia sesión en el sistema.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Accede a la página de login.	2. Pide al usuario sus credenciales: correo electrónico y contraseña.
3. Introduce sus credenciales y las confirma	4. Pide al servidor si el usuario con las credenciales introducidas existe. 5. Redirecciona al usuario a la página principal y lo inicia en sesión
Flujo de acciones de caminos alternativos	
El usuario no está registrado.	
La contraseña introducida es inválida.	

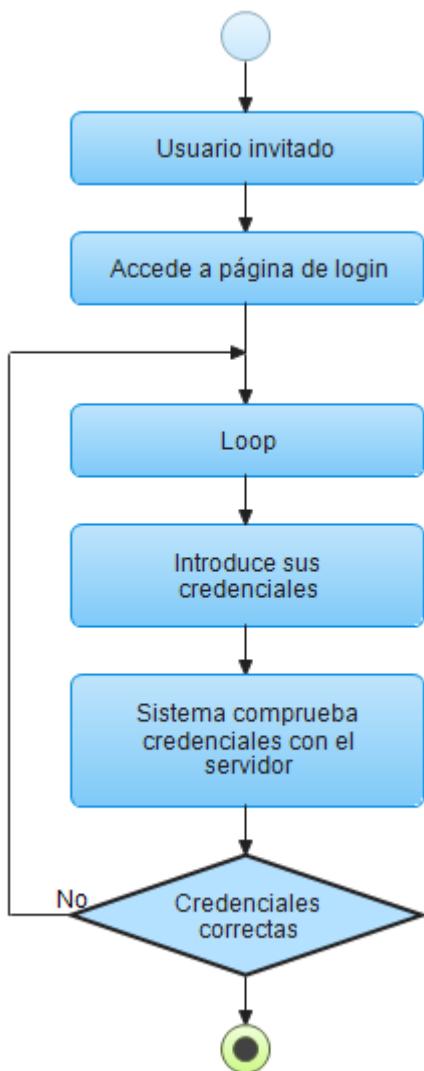


Figura 24 Diagrama de actividad - CU-W-2 Iniciar sesión

4.4.2.3 CU-W-3 Ver actividades

Caso de uso	Ver actividades
Actores	Usuario invitado, Usuario identificado
Iniciado por	Usuario invitado, Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	Debe haber algún viaje registrado y pendiente de realización en el sistema
Postcondiciones	-
Requisitos que satisface	RFW3, RFW4
Propósito	
Ver la lista de actividades pendientes de realización con la información básica de cada una de ellas.	
Resumen de alto nivel	
El usuario accede a la página principal del sistema donde ve la lista de actividades pendientes de realización pudiendo filtrarlas para obtener resultados diferentes	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la página principal del sistema	2. Se recuperan de la base de datos las últimas actividades pendientes de realización 3. Devuelve la lista de las actividades
4. Introduce criterios de búsqueda para actividades	5. Filtra las actividades por el criterio de búsqueda introducido 6. Devuelve la lista de actividades
Flujo de acciones de caminos alternativos	
El usuario previamente hace login en el sistema y accede a la página principal donde ve la lista de viajes.	
El usuario introduce un filtro para cambiar los viajes a visualizar.	
El usuario accede al sistema y en la lista de actividades filtra por actividades realizadas	

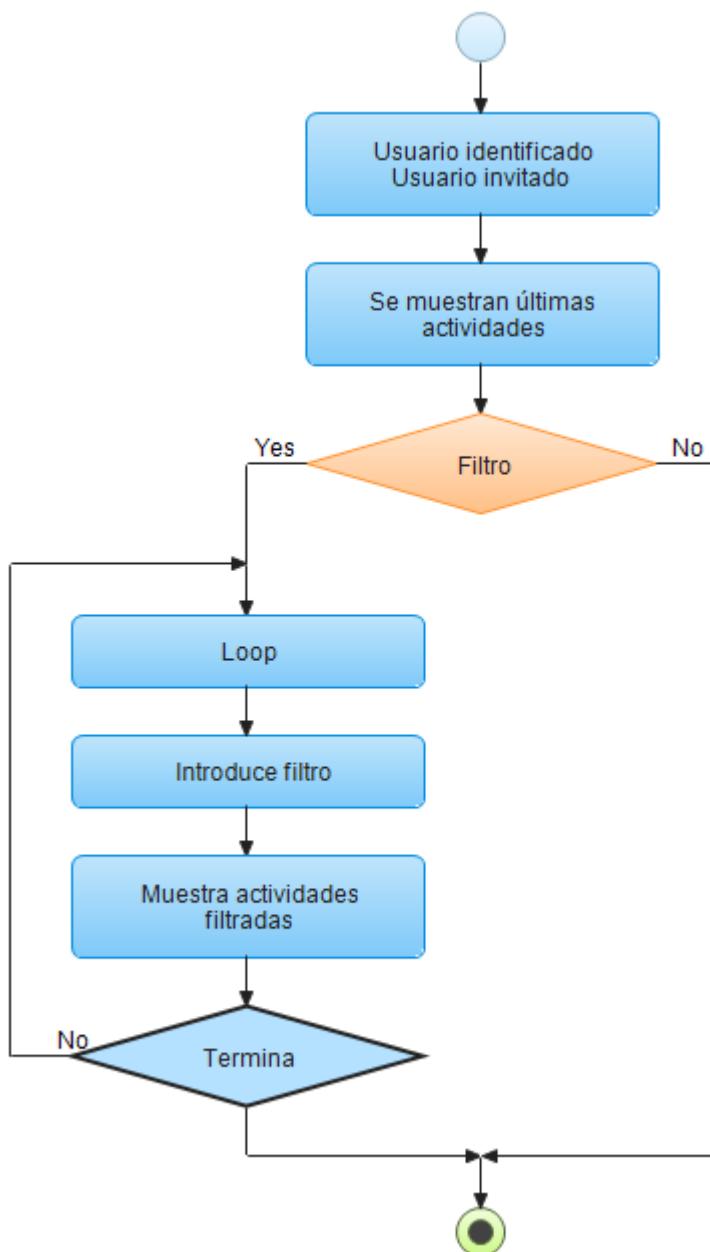


Figura 25 Diagrama de actividad - CU-W-3 Ver actividades

4.4.2.4 CU-W-4 Recuperar cuenta de usuario

Caso de uso	Recuperar cuenta de usuario
Actores	Usuario invitado
Iniciado por	Usuario
Tipo	Secundario
Referencias	-
Precondiciones	El usuario no deberá estar identificado en el sistema.
Postcondiciones	Una vez ejecutado el caso de uso el usuario tendrá una nueva password asociada a la cuenta.
Requisitos que satisface	RFW6
Propósito	Cambiar la contraseña aportando el correo electrónico

Resumen de alto nivel	
El usuario a través del correo electrónico solicita una nueva contraseña para su cuenta	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario desde la pantalla de login solicita una nueva contraseña para su cuenta introduciendo el correo electrónico	2. Genera una nueva contraseña aleatoria y la asigna a la cuenta 3. Envía un correo electrónico al usuario con la nueva contraseña generada.
Flujo de acciones de caminos alternativos	
El correo electrónico aportado no coincide con ninguna cuenta de usuario registrada en la aplicación.	

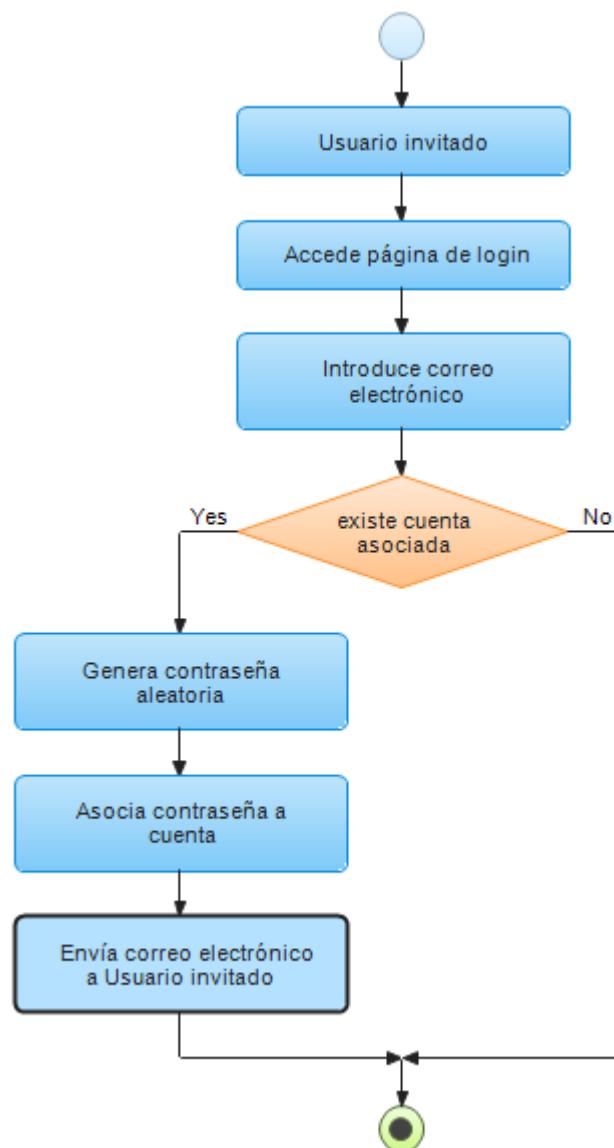


Figura 26 Diagrama de actividad - CU-W-4 Recuperar cuenta de usuario

4.4.2.5 CU-W-5 Cerrar sesión

Caso de uso	Cerrar sesión	
Actores	Usuario identificado	
Iniciado por	Usuario	
Tipo	Secundario	
Referencias	-	
Precondiciones	El usuario deberá estar previamente logueado en el sistema.	
Postcondiciones	El usuario después de la ejecución del caso de uso no debería estar logueado en el sistema.	
Requisitos que satisface	RFW13	
Propósito	Cerrar sesión de la página web	
Cerrar sesión de la página web	Resumen de alto nivel	
Resumen de alto nivel	El usuario con sesión iniciada cierra sesión	
El usuario con sesión iniciada cierra sesión	Descripción detallada	
Descripción detallada	Flujo de acciones del camino básico	
Flujo de acciones del camino básico	ACTOR	SISTEMA
1. El usuario desde cualquier parte de la aplicación cierra sesión en el sistema	2. Cierra la sesión del usuario. 3. Redirige al usuario a la página de login.	
Flujo de acciones de caminos alternativos	-	
-		

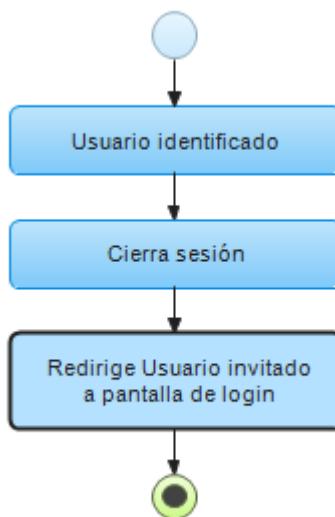


Figura 27 Diagrama de actividad – CU-W-5 cerrar sesión

4.4.2.6 CU-W-6 Borrar cuenta

Caso de uso	Borrar cuenta
Actores	Usuario identificado
Iniciado por	Usuario
Tipo	Secundario
Referencias	-
Precondiciones	El usuario deberá estar previamente logueado en el sistema.

Postcondiciones	Después de la ejecución del caso de uso el usuario estará deslogueado y la cuenta eliminada
Requisitos que satisface	RFW14
Propósito	
Realizar un borrado lógico de la cuenta	
Resumen de alto nivel	
El usuario con sesión iniciada elimina la cuenta	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a su perfil en la página web y hace click sobre el enlace a eliminar la cuenta 2. Confirma el diálogo emergente de confirmación de borrado	3. Realiza un borrado lógico de la cuenta de usuario 4. Cierra sesión al usuario
Flujo de acciones de caminos alternativos	
El usuario no confirma la eliminación de la cuenta	

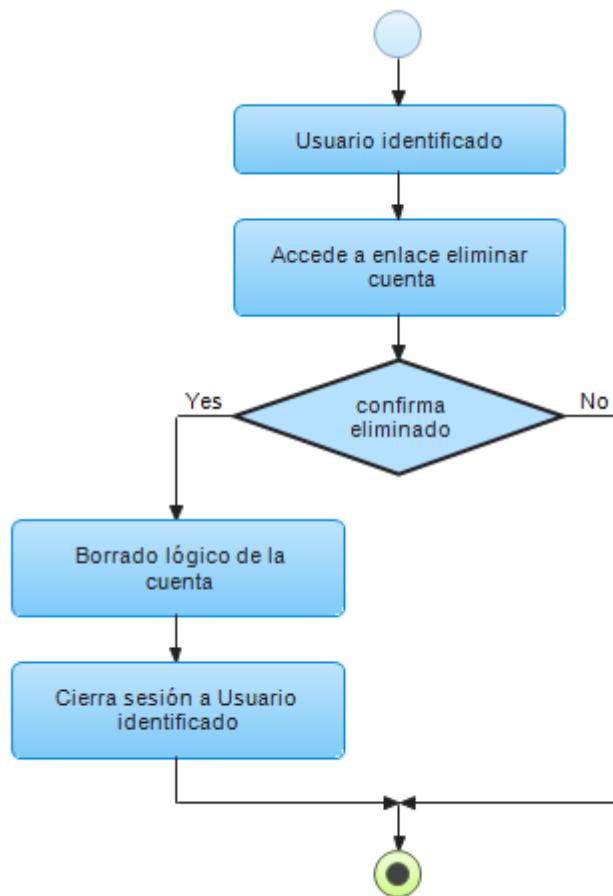


Figura 28 Diagrama de actividad - CU-W-6 Borrar cuenta

4.4.2.7 CU-W-7 Crear actividad

Caso de uso	Crear actividad
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	El usuario deberá estar previamente logueado en el sistema.
Postcondiciones	Después de la ejecución del caso de uso deberá existir una nueva actividad identificada de forma única y con los datos aportados por el usuario, y además está disponible para verla desde la página principal de la aplicación.
Requisitos que satisface	RFW9
Propósito	
Crear una nueva actividad en el sistema	
Resumen de alto nivel	
El usuario accede al sistema y aportando los datos de la actividad crea una nueva.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario introduce una breve descripción de la actividad, una descripción completa de la ruta a seguir o bien la URL que contenga el detalle de la misma, fecha y hora de comienzo, plazas máximas.	2. El sistema valida que todos los datos hayan sido introducidos y correctos: la fecha introducida ha de ser posterior a la actual.
3. El usuario carga desde su ordenador local el archivo con extensión .gpx que contiene los datos del track de la ruta.	4. El sistema valida que el fichero tenga la extensión requerida (.gpx) 5. Procesa los datos del .gpx
6. El usuario selecciona el/los correos electrónicos de el/los guía/s de la actividad.	7. El sistema verifica que al menos un usuario ha sido seleccionado como guía para la actividad.
8. El usuario confirma los datos introducidos	9. El sistema envía al servidor los datos de la nueva actividad, así como el fichero. gpx con los datos de la ruta. 10. El sistema redirige al usuario al detalle de la actividad
Flujo de acciones de caminos alternativos	
El track de la ruta introducido es inválido.	

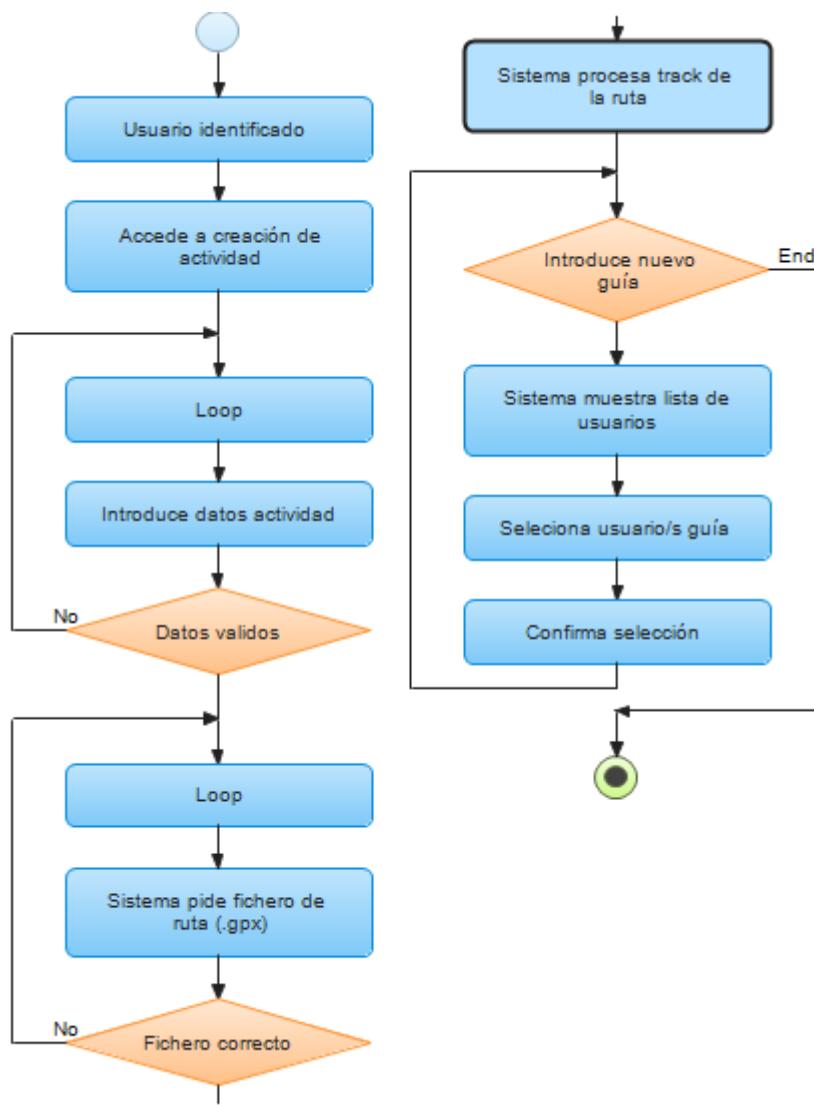


Figura 29 Diagrama de actividad - CU-W-7 Crear actividad

4.4.2.8 CU-W-8 Editar actividad

Caso de uso	Editar actividad
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Secundario
Referencias	-
Precondiciones	El usuario deberá estar previamente identificado en el sistema. El usuario deberá tener al menos una actividad creada por él mismo. La actividad no deberá haber sido finalizada ya.
Postcondiciones	Después de la ejecución del caso de uso la actividad deberá tener los datos actualizados con los nuevos introducidos por el usuario.
Requisitos que satisface	RFW10
Propósito	

Editar los datos de una actividad existente en el sistema.	
Resumen de alto nivel	
El usuario accede al sistema, accede a sus actividades, selecciona la que quiere editar, cambia los campos oportunos y confirma los cambios.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la actividad que quiere editar	2. El sistema descarga la contraseña de la actividad y se la solicita al usuario
3. El usuario introduce la contraseña de la actividad y la envía.	4. El sistema verifica que la contraseña coincida con la de la actividad. 5. El sistema descarga y muestra al usuario los datos de la actividad
6. El usuario accede a la edición de la actividad	7. El sistema muestra al usuario los campos editables de la información de la actividad
8. El usuario edita la actividad y confirma los cambios	9. El sistema guarda la información nueva para la actividad en la base de datos. 10. Redirecciona al usuario a la página de la actividad.
Flujo de acciones de caminos alternativos	
La contraseña que aporta el usuario para la actividad es incorrecta. El track de la ruta introducido nuevo es inválido. La actividad ya está cerrada tras haber sido finalizada. Alguno de los datos obligatorios para la actividad no es rellenado.	

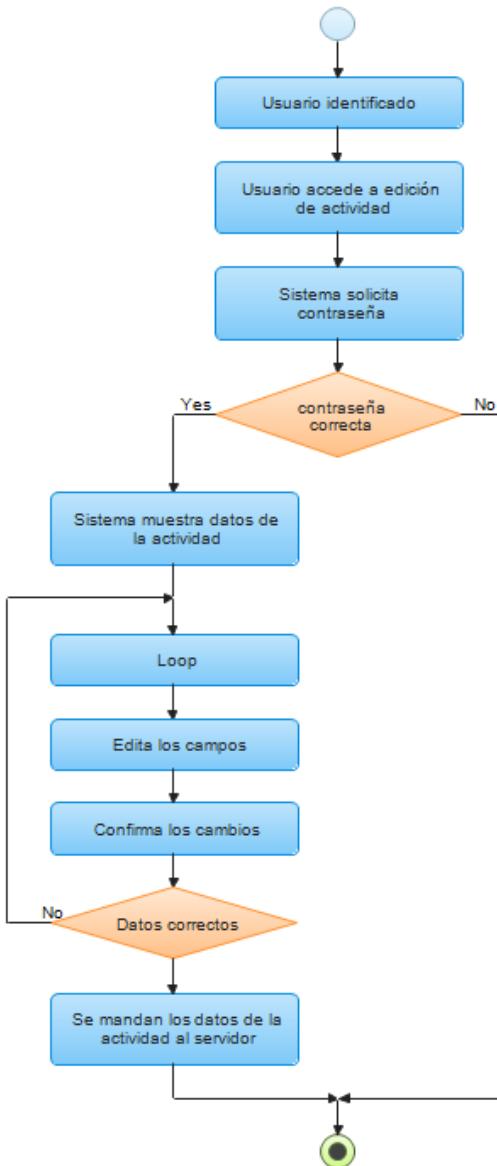


Figura 30 Diagrama de actividad - CU-W-8 Editar actividad

4.4.2.9 CU-W-9 Cambiar datos personales

Caso de uso	Cambiar datos personales
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Secundario
Referencias	-
Precondiciones	El usuario deberá estar previamente identificado en el sistema.
Postcondiciones	Tras la ejecución del caso de uso el usuario deberá tener sus datos actualizados acorde a la nueva información aportada.
Requisitos que satisface	RFW7
Propósito	Editar los datos personales de un usuario registrado en el sistema.
Resumen de alto nivel	

El usuario accede al sistema, edita sus datos personales	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la página de datos personales	2. Descarga todos los datos personales del usuario y los muestra para su edición.
3. El usuario hace cambios en sus datos y los confirma	4. Valida los datos introducidos. 5. Envía al servidor los datos del usuario 6. Redirecciona al usuario a la página de su perfil.
Flujo de acciones de caminos alternativos	
Las contraseñas nuevas no coinciden entre ellas. El correo electrónico nuevo aportado no es válido. El correo electrónico nuevo aportado ya está en uso.	

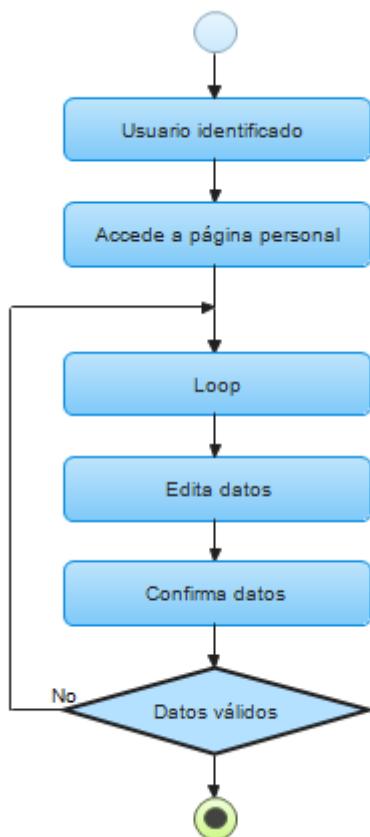


Figura 31 Diagrama de actividad – CU-W-10 Editar datos personales

4.4.2.10 CU-W-10 Monitorizar actividad

Caso de uso	Monitorizar actividad
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	Debe haber una actividad en curso en el sistema.

	Deben haber personas siendo monitorizadas en la actividad.
Postcondiciones	-
Requisitos que satisface	RFW12
Propósito	
Conocer la posición en tiempo real de las personas que están siendo monitorizadas en una actividad, así como el track previsto de la ruta.	
Resumen de alto nivel	
El usuario accede al sistema, accede a una actividad que esté en curso, introduce la contraseña de la misma, y ve la información de la posición de los excursionistas, así como el track de la ruta.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la página de una actividad	2. El sistema recupera los datos de la actividad y se los muestra al usuario
3. El usuario accede a la página de monitorización de una actividad e introduce la contraseña de la misma.	4. El sistema comprueba que la contraseña introducida para la actividad es correcta 5. Descarga los datos de posición de los excursionistas de la actividad.
Flujo de acciones de caminos alternativos	
La contraseña para la actividad es inválida. No existen datos de monitorización aún en la aplicación.	

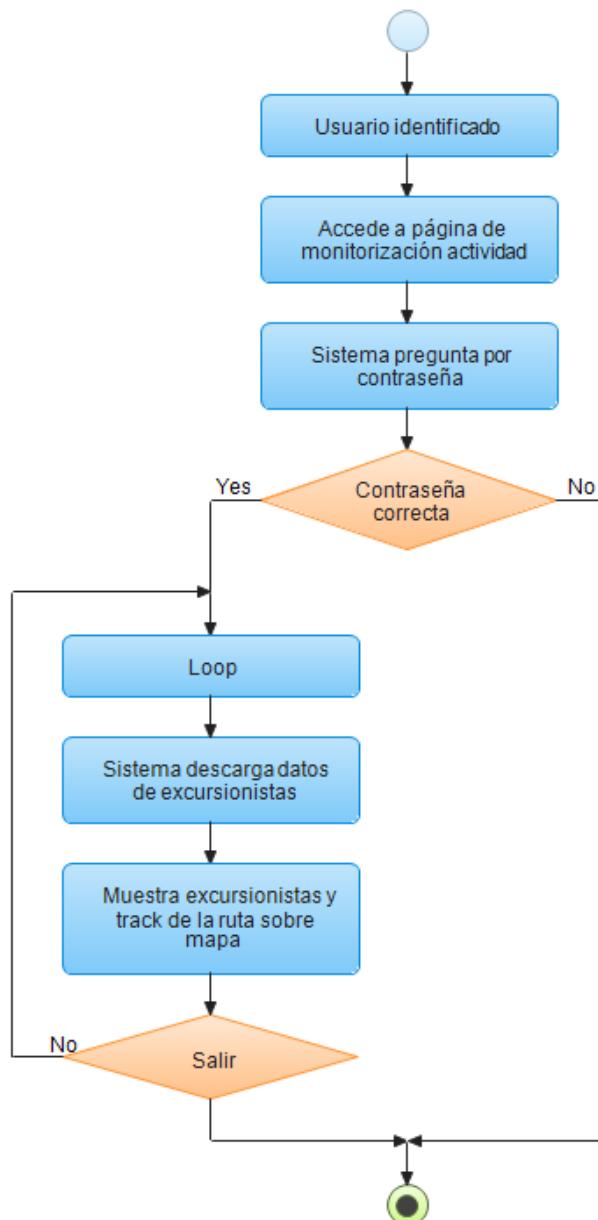


Figura 32 Diagrama de actividad - CU-W-10 Monitorizar actividad

4.4.2.11 CU-W-11 Eliminar actividad

Caso de uso	Eliminar actividad
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	Debe haber una actividad en estado PENDING en el sistema creada por el usuario que ejecuta el caso de uso.
Postcondiciones	-
Requisitos que satisface	RFW11
Propósito	Realizar un borrado lógico de una actividad

Resumen de alto nivel	
El usuario accede a la vista de la actividad, accede al enlace para eliminar la actividad, confirma la acción y se marca como eliminada.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la página de una actividad 3. Accede al enlace para eliminar la actividad 4. Confirma la acción	2. El sistema recupera los datos de la actividad y se los muestra al usuario 5. Marca la actividad como eliminada 6. Redirecciona al usuario a la lista de actividades pendientes
Flujo de acciones de caminos alternativos	
El usuario no es el organizador de la actividad El usuario no confirma la acción de eliminar la actividad La actividad no se encuentra en estado PENDING	

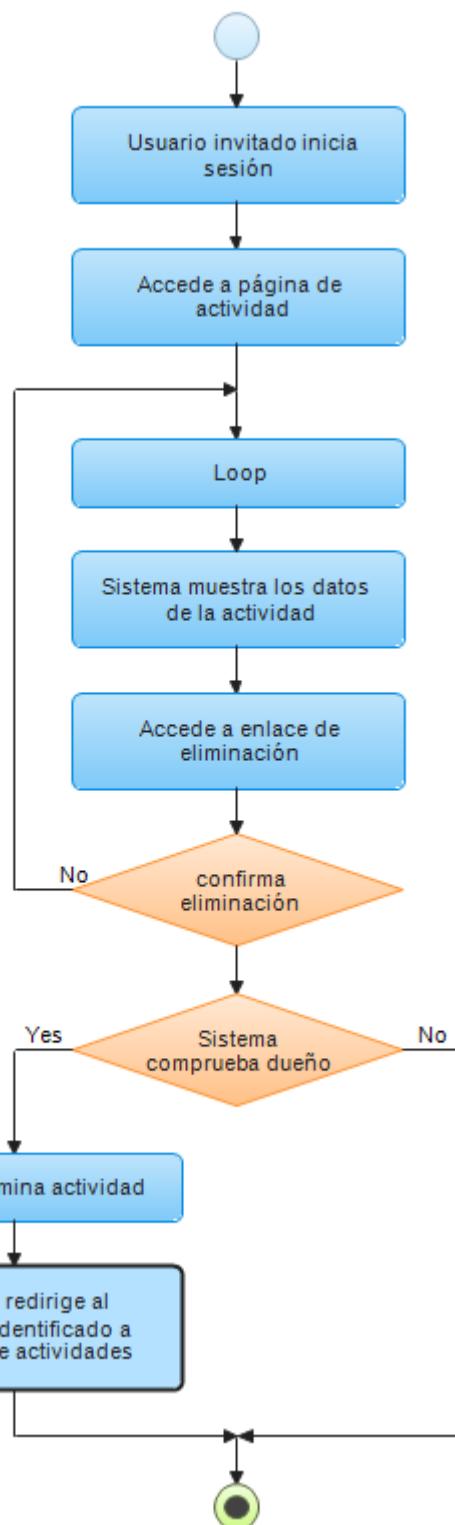


Figura 33 Eliminar actividad

4.4.3 Especificación de casos de uso y escenarios subsistema aplicación móvil

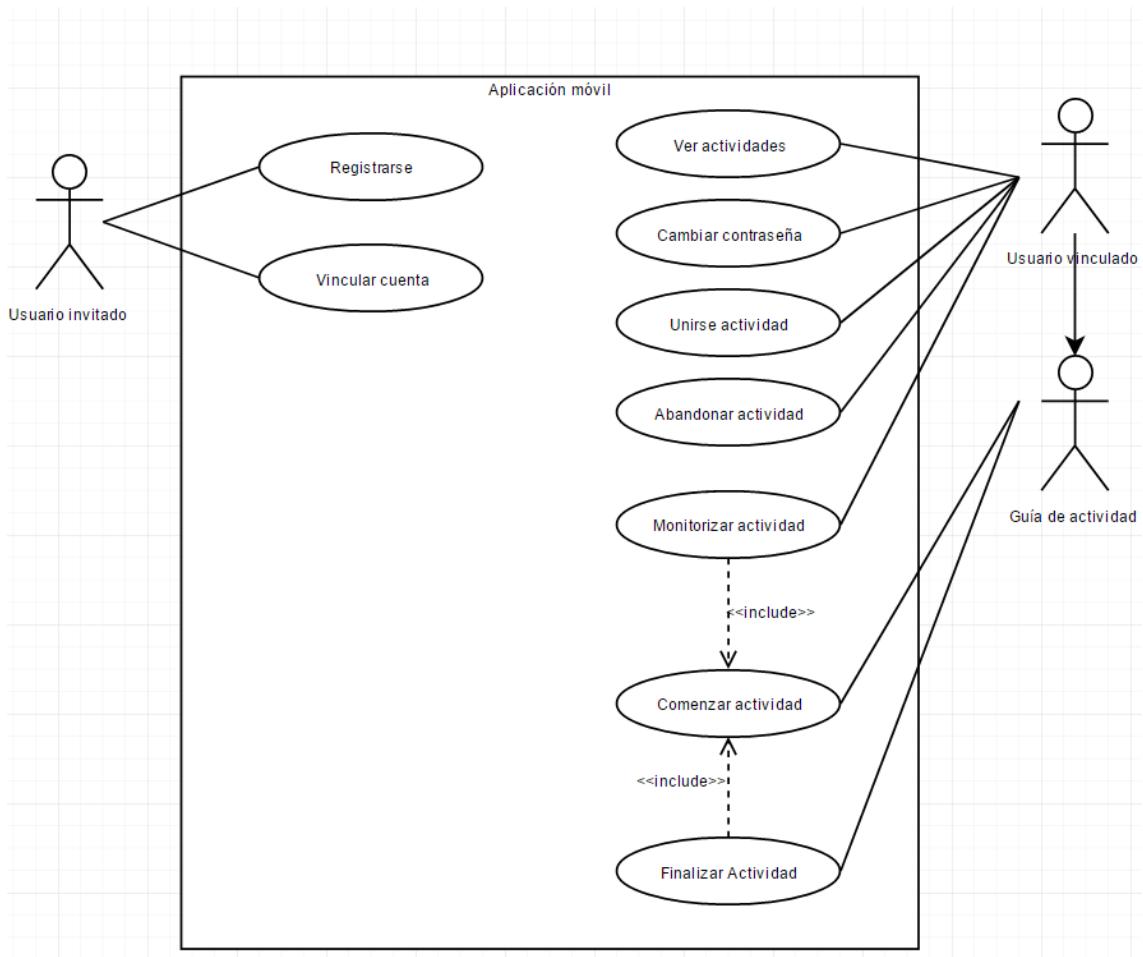


Figura 34 Diagrama de casos de uso subsistema aplicación móvil

Caso de uso	Descripción
Registrarse	Un usuario invitado podrá registrarse en la aplicación aportando para ello el correo electrónico, nombre apellidos y contraseña.
Vincular cuenta	Vinculación de la cuenta con la aplicación por medio de las credenciales.
Ver actividades	Ver la lista de actividades pendientes de realización registradas en el sistema
Cambiar contraseña	Actualizar la contraseña de la cuenta aportando la actual
Unirse actividad	Unirse como participante (monitorizado) a una actividad que esté en curso en el sistema.
Abandonar actividad	Abandonar una actividad y por tanto dejar de ser monitorizado
Monitorizar actividad	Monitorización de una actividad, ver toda la información de las personas que son monitorizados, así como su posición a tiempo real y el track de la ruta.
Comenzar actividad	Dar por comenzada una actividad para poder ser monitorizado en la misma.
Finalizar actividad	Dar por finalizada una actividad por la cual se dejará de monitorizar a los participantes.

4.4.3.1 CU-M-1 Registrarse

Caso de uso	Registrarse
Actores	Usuario invitado
Iniciado por	Usuario invitado
Tipo	Primario
Referencias	-
Precondiciones	El usuario no debe tener su cuenta vinculada.
Postcondiciones	Tras la ejecución del caso de uso el usuario deberá estar registrado en el sistema
Requisitos que satisface	RFM1
Propósito	Registrar una nueva cuenta de usuario en el sistema.
Resumen de alto nivel	El usuario accede a la aplicación e introduce sus datos de registro. En caso de ser correctos es registrado y vinculado con el sistema.
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. El usuario accede a la aplicación e introduce su contraseña (y repetida), su correo electrónico, su nombre y sus apellidos.	2. El sistema valida los datos introducidos por el usuario: comprueba que las contraseñas coincidan entre si y cumplan el formato requerido, así como el correo electrónico. 3. Se valida que el correo electrónico no esté en uso por otro usuario registrado en el sistema.
4. Confirma y envía sus datos	5. El sistema envía los datos del nuevo usuario al servidor. 6. Se vincula la cuenta con la aplicación.
Flujo de acciones de caminos alternativos	
El usuario ya tiene su cuenta vinculada en la aplicación. Las contraseñas del registro no coinciden entre ellas. El formato de contraseña no es el requerido. El correo ya está en uso en el sistema.	

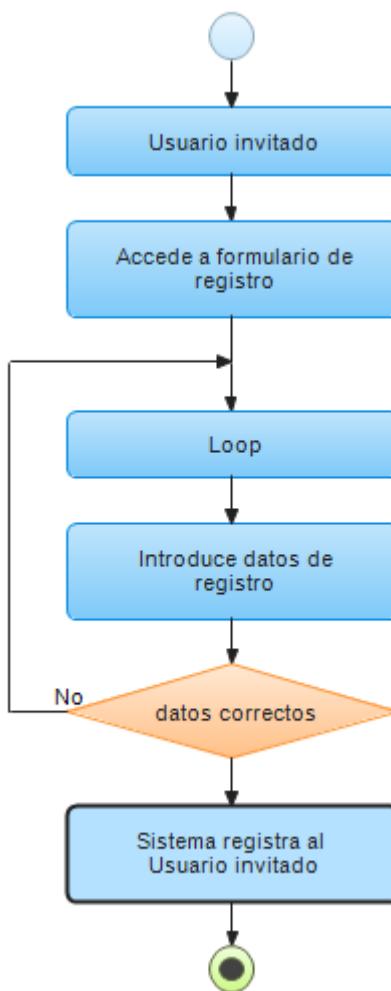


Figura 35 Diagrama de actividad - CU-M-1 Registrarse

4.4.3.2 CU-M-2 Vincular cuenta

Caso de uso	Vincular cuenta
Actores	Usuario invitado
Iniciado por	Usuario invitado
Tipo	Primario
Referencias	-
Precondiciones	El usuario no debe tener su cuenta vinculada. El usuario deberá estar registrado en el sistema.
Postcondiciones	La cuenta del usuario deberá estar vinculada con el sistema.
Requisitos que satisface	RFM2, RFM3
Propósito	Vincular una cuenta de usuario con la aplicación para que no sea necesario identificarse en el futuro usando el mismo dispositivo.
Resumen de alto nivel	El usuario accede a la aplicación e introduce sus credenciales de acceso.
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA

1. El usuario accede a la aplicación e introduce su correo electrónico y contraseña.	2. La aplicación envía al servidor los datos introducidos por el usuario y comprueba si existe un usuario con las credenciales aportadas. 3. La aplicación vincula la cuenta del usuario
Flujo de acciones de caminos alternativos	
El usuario ya tiene una cuenta vinculada en el sistema. Las credenciales que se introducen son inválidas.	
El usuario ya tiene una cuenta vinculada y no tiene conexión a internet, en cuyo caso iniciaría también en sesión.	

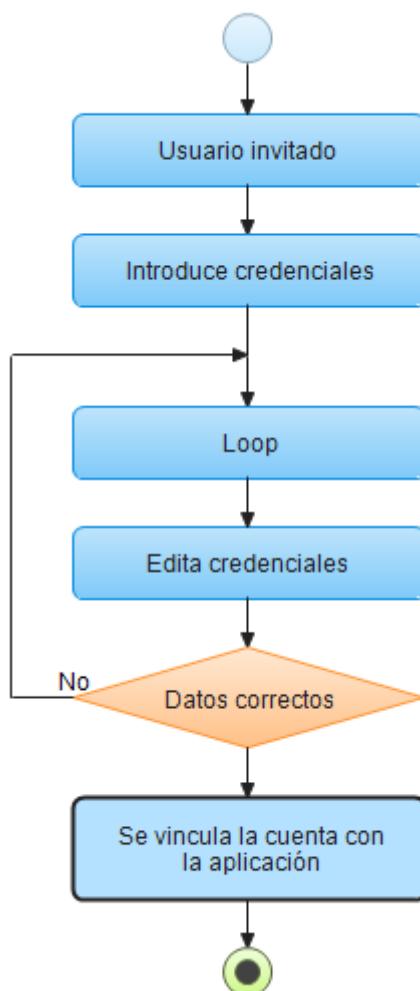


Figura 36 Diagrama de actividad – CU-M-2 Vincular cuenta

4.4.3.3 CU-M-3 Ver actividades

Caso de uso	Ver actividades
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	El usuario debe tener su cuenta vinculada en el sistema.

Postcondiciones	-
Requisitos que satisface	RFM4, RFM5
Propósito	
Ver las actividades pendientes de realización.	
Resumen de alto nivel	
El usuario accede a la aplicación y descarga la lista de actividades pendientes de realización	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Usuario accede a la aplicación	2. La aplicación solicita al servidor la lista de actividades pendientes de realización. 3. La aplicación muestra al usuario las actividades.
Flujo de acciones de caminos alternativos	
No hay actividades pendientes de realización en el sistema.	
No tiene conexión a internet, mostraría las actividades guardadas en el dispositivo	

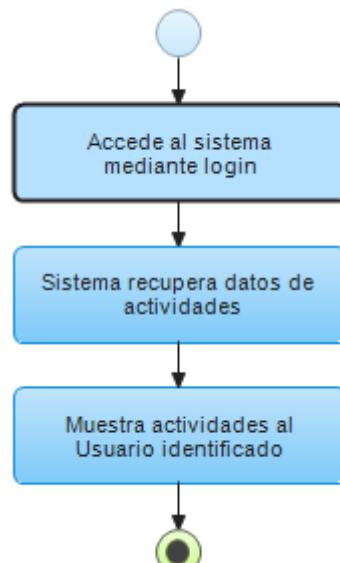


Figura 37 Diagrama de actividad - CU-M-3 Ver actividades

4.4.3.4 CU-M-4 Comenzar actividad

Caso de uso	Comenzar actividad
Actores	Guía de la actividad
Iniciado por	Guía de la actividad
Tipo	Primario
Referencias	-
Precondiciones	Debe de haber una actividad pendiente de realización en el sistema. El usuario debe ser uno de los guías de la actividad pendiente de realización.
Postcondiciones	Tras la ejecución del caso de uso el estado de la actividad debe ser en curso.
Requisitos que satisface	RFM6

Propósito	
Dar una actividad por comenzada.	
Resumen de alto nivel	
Se accede a la aplicación y a la actividad que se quiere iniciar y se introduce una contraseña para la misma.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Introduce la contraseña para la actividad	2. Envía al sistema la contraseña introducida y cambia el estado de la actividad a en curso.
Flujo de acciones de caminos alternativos	
No se introduce una contraseña válida para la actividad.	

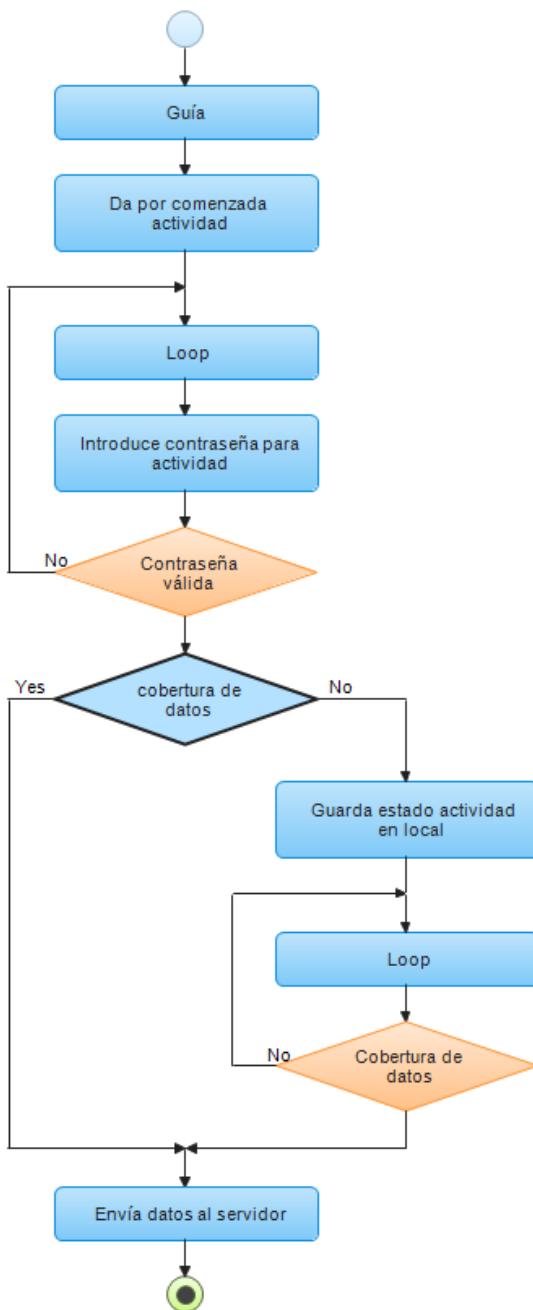


Figura 38 Diagrama de actividad – CU-M-4 Comenzar actividad

4.4.3.5 CU-M-5 Monitorizar actividad

Caso de uso	Monitorizar actividad
Actores	Guía de la actividad, Usuario identificado
Iniciado por	Guía de la actividad o Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	Debe haber una actividad en curso en el sistema. El usuario deberá estar siendo monitorizado.
Postcondiciones	-
Requisitos que satisface	RFM9, RFM11

Propósito	
Conocer la posición de todos los excursionistas de una actividad, así como el track esperado para la ruta.	
Resumen de alto nivel	
Se accede a la monitorización de la actividad en la que se está siendo monitorizado.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Accede a la actividad en la que está siendo monitorizado	2. Descarga los datos del track de la ruta y la posición del resto de participantes en la actividad.
Flujo de acciones de caminos alternativos	
No se introduce una contraseña válida para la actividad.	

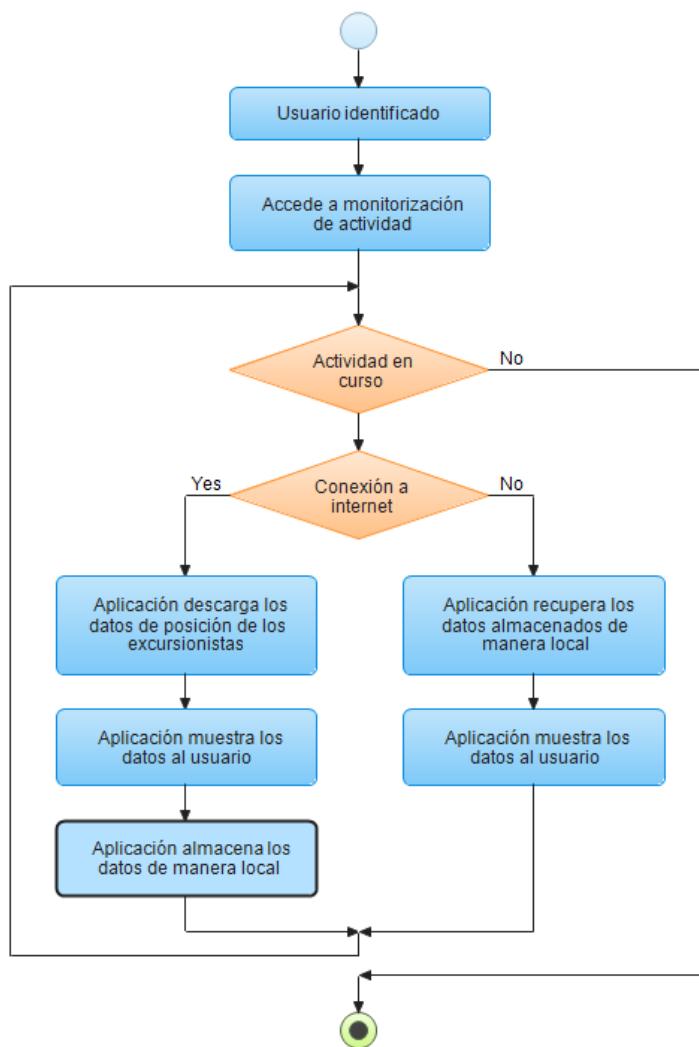


Figura 39 Diagrama de actividad – CU-M-5 Monitorizar actividad aplicación móvil

4.4.3.6 CU-M-6 Unirse a actividad

Caso de uso	Unirse a actividad
--------------------	--------------------

Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	Debe haber una actividad en curso en el sistema. El usuario no deberá estar ya siendo monitorizado en alguna actividad.
Postcondiciones	El sistema deberá conocer la posición del usuario y deberá figurar como excursionista de la actividad.
Requisitos que satisface	RFM7
Propósito	
Participar como usuario monitorizado en una actividad para que recoja de forma periódica los datos de posición.	
Resumen de alto nivel	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Accede a la actividad a la que quiere unirse. 2. Se confirma como participante de la actividad	3. Envía al servidor los datos del nuevo participante. 4. Descarga los datos de posición del usuario de forma periódica. 5. Envía datos al servidor de la posición del usuario
Flujo de acciones de caminos alternativos	
No hay cobertura para realizar la operación.	

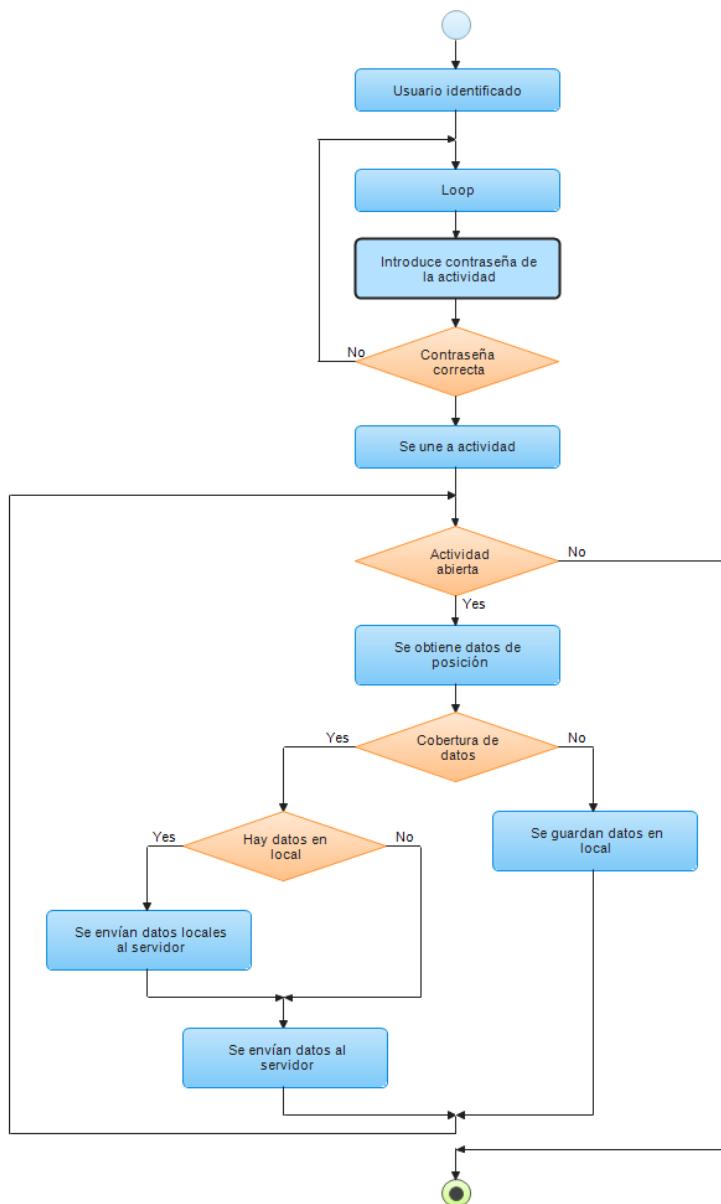


Figura 40 Diagrama de actividad – CU-M-6 Unirse a actividad

4.4.3.7 CU-M-7 Finalizar actividad

Caso de uso	Finalizar actividad
Actores	Guía actividad
Iniciado por	Guía actividad
Tipo	Primario
Referencias	-
Precondiciones	Debe haber una actividad en curso en el sistema. El usuario deberá ser guía de la actividad
Postcondiciones	La actividad deberá estar en estado finalizada.
Requisitos que satisface	RFM12
Propósito	Dar por finalizada una actividad
Resumen de alto nivel	

El guía accede a la actividad en curso y la da por finalizada.	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Accede a la actividad a la que quiere dar por finalizada.	2. Actualiza en el servidor el nuevo estado de la actividad.
Flujo de acciones de caminos alternativos	
No hay cobertura para realizar la operación.	

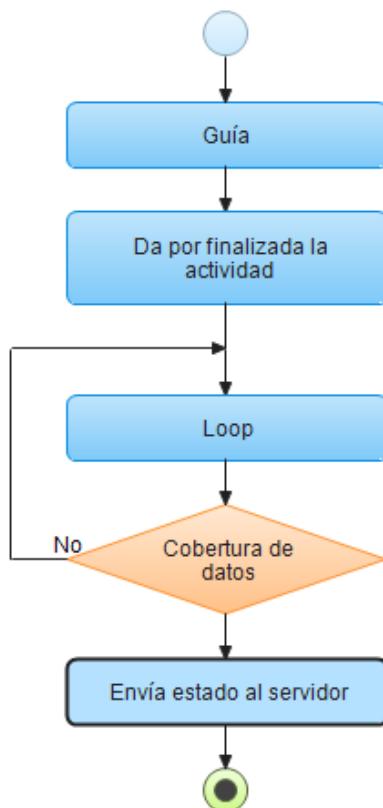


Figura 41 Diagrama de actividad – CU-M-7 Finalizar actividad

4.4.3.8 CU-M-8 Cambiar password

Caso de uso	Cambiar password
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Secundario
Referencias	-
Precondiciones	El usuario debe estar identificado en el sistema
Postcondiciones	El usuario tendrá que tener la nueva password actualizada
Requisitos que satisface	
RFM8	
Propósito	

Cambiar la password de un usuario								
Resumen de alto nivel								
El usuario accede a la vista de datos personales y cambia su contraseña.								
Descripción detallada								
Flujo de acciones del camino básico								
<table border="1"> <thead> <tr> <th>ACTOR</th> <th>SISTEMA</th> </tr> </thead> <tbody> <tr> <td>1. Accede a la actividad de datos personales</td> <td>2. Recupera su nombre de usuario, correo electrónico y lo muestra</td></tr> <tr> <td>3. Introduce una nueva password 'asd' y la de confirmación 'asd'</td> <td>4. Valida que las password introducidas coincidan 5. Actualiza la password al usuario 6. Notifica al usuario</td></tr> <tr> <td></td> <td></td></tr> </tbody> </table>	ACTOR	SISTEMA	1. Accede a la actividad de datos personales	2. Recupera su nombre de usuario, correo electrónico y lo muestra	3. Introduce una nueva password 'asd' y la de confirmación 'asd'	4. Valida que las password introducidas coincidan 5. Actualiza la password al usuario 6. Notifica al usuario		
ACTOR	SISTEMA							
1. Accede a la actividad de datos personales	2. Recupera su nombre de usuario, correo electrónico y lo muestra							
3. Introduce una nueva password 'asd' y la de confirmación 'asd'	4. Valida que las password introducidas coincidan 5. Actualiza la password al usuario 6. Notifica al usuario							
Flujo de acciones de caminos alternativos								
Las passwords isntradicadas por el usuario no coinciden.								

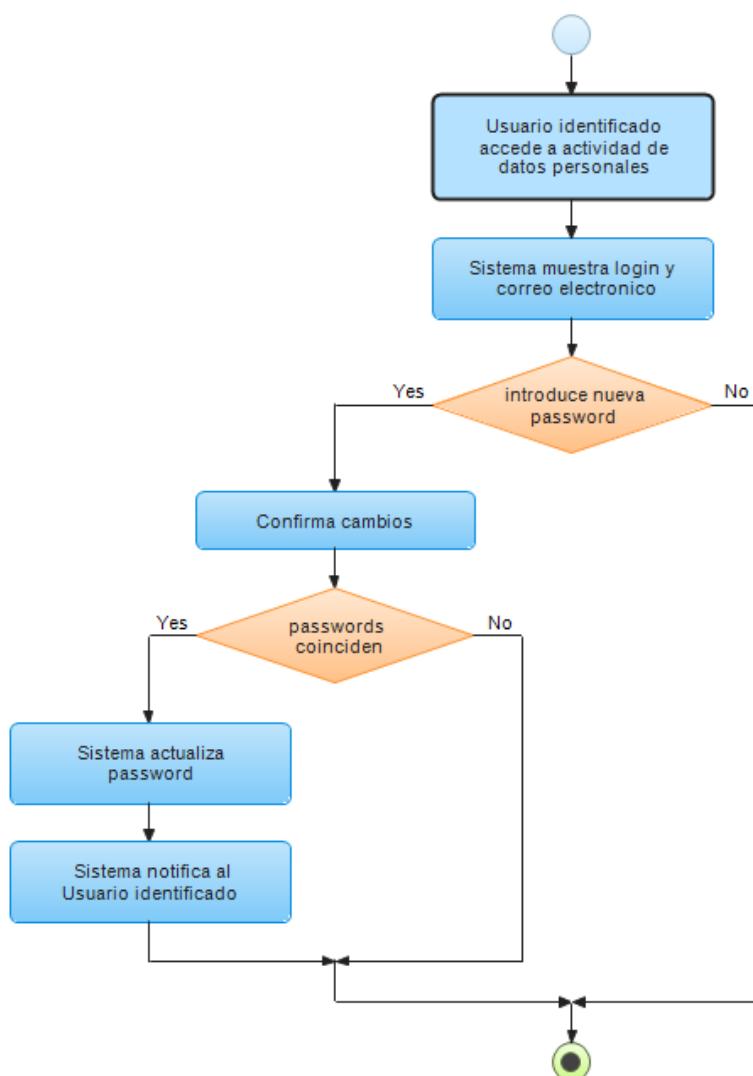


Figura 42 Diagrama de actividad - CU-M-8 Cambiar password

4.4.3.9 CU-M-9 Abandonar actividad

Caso de uso	Abandonar actividad
92	Víctor Castaño Gutiérrez Escuela de Ingeniería Informática - Universidad de Oviedo

Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Primario
Referencias	-
Precondiciones	El usuario debe estar identificado en el sistema El usuario debe estar unido a una actividad en curso
Postcondiciones	El usuario no estará unido en ninguna actividad en curso El usuario no estará siendo monitorizado
Requisitos que satisface	RFM13
Propósito	
Abandonar una actividad que está en curso y detener el proceso de monitorización	
Resumen de alto nivel	
El usuario accede a la vista de la actividad y detiene la monitorización abandonando dicha actividad	
Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Accede a la vista de la actividad que está en curso en la que está unido	2. Recupera los datos de la actividad y los muestra al usuario. 3. Habilita el acceso a abandonar la actividad
3. Confirma la intención de abandonar la actividad	4. Valida que el usuario esté en la actividad. 5. Desvincula al usuario de la actividad 6. Detiene el proceso de monitorización
Flujo de acciones de caminos alternativos	
-	

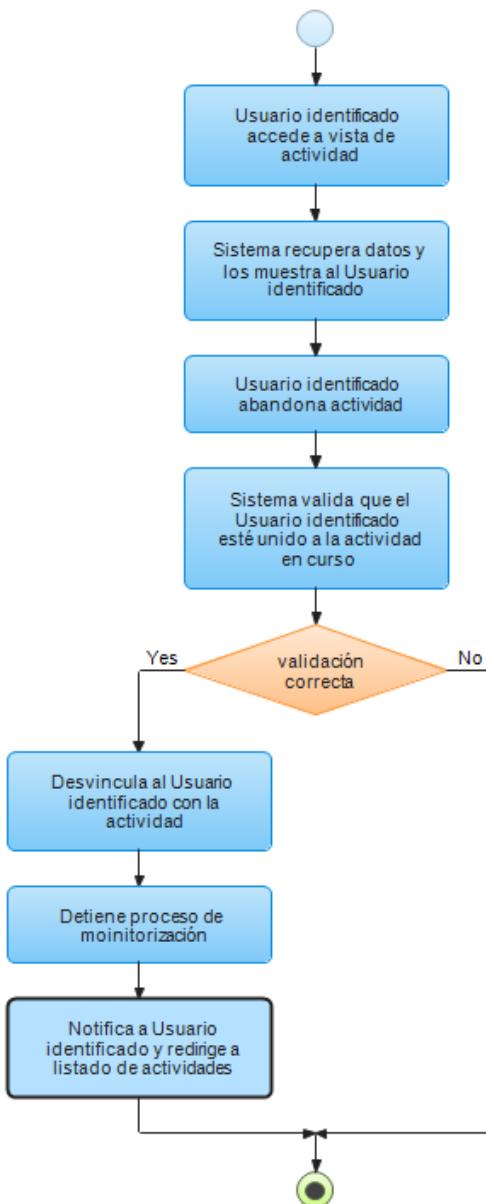


Figura 43 Diagrama de actividad - Abandonar actividad

4.4.3.10 CU-M-10 Cerrar sesión

Caso de uso	Cerrar sesión
Actores	Usuario identificado
Iniciado por	Usuario identificado
Tipo	Secundario
Referencias	-
Precondiciones	El usuario debe estar identificado en el sistema
Postcondiciones	El usuario no estará identificado en el sistema El usuario no tendrá datos de acceso de sesión guardados
Requisitos que satisface	RFM10
Propósito	Cerrar sesión estando identificado y eliminar los datos de inicio de sesión automático
Resumen de alto nivel	El usuario cierra sesión de la aplicación estando previamente identificado.

Descripción detallada	
Flujo de acciones del camino básico	
ACTOR	SISTEMA
1. Cierra sesión	2. Cierra sesión al usuario 3. Elimina datos de acceso de sesión 4. Redirige a vista de login
Flujo de acciones de caminos alternativos	
-	

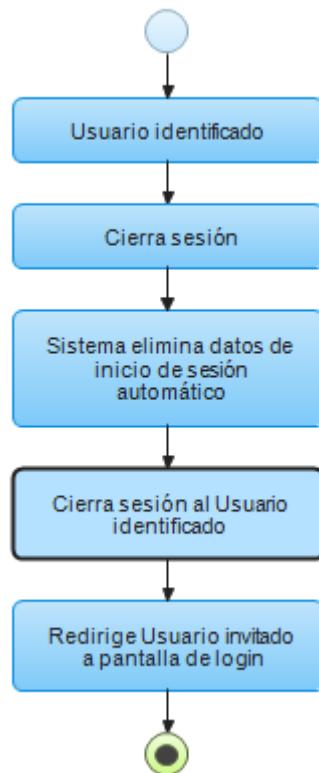


Figura 44 Diagrama de actividad - Cerrar sesión

4.4.4 Matriz de trazabilidad requisitos funcionales - casos de uso subsistema web

	CU-W-1 Registrarse	CU-W-2 Iniciar sesión	CU-W-3 Ver actividades	CU-W-4 Recuperar cuenta	CU-W-5 Cerrar sesión	CU-W-6 Borrar cuenta	CU-W-7 Crear actividad	CU-W8 Editar actividad	CU-W-9 Cambiar datos personales	CU-W-10 Monitorizar actividad	CU-W-11 Eliminar actividad
RFW1	✓										
RFW2		✓									
RFW3			✓								
RFW4				✓							
RFW5									✓		
RFW6					✓						
RFW7									✓		
RFW8							✓	✓		✓	✓
RFW9							✓				
RFW10								✓			
RFW11											✓
RFW12										✓	
RFW13					✓						
RFW14						✓					

Tabla 12 Matriz de trazabilidad requisitos funcionales con casos de uso – subsistema web

4.4.5 Matriz de trazabilidad requisitos funcionales – casos de uso subsistema aplicación móvil

	CU-M-1 Registrarse	CU-M-2 Vincular cuenta	CU-M-3 Ver actividades	CU-M-4 Comenzar actividad	CU-M-5 Monitorizar actividad	CU-M-6 Unirse actividad	CU-M-7 Finalizar actividad	CU-M-8 Cambiar password	CU-M-9 Abandonar actividad	CU-M-10 Cerrar sesión
RFM1	✓									
RFM2		✓								
RFM3		✓								
RFM4			✓							
RFM5			✓							
RFM6				✓						
RFM7						✓				
RFM8								✓		
RFM9					✓					
RFM10										✓
RFM11					✓					
RFM12							✓			
RFM13									✓	

Tabla 13 Matriz de trazabilidad requisitos funcionales con casos de uso - subsistema móvil

4.5 Análisis de Interfaces de Usuario

En este punto de la fase de análisis, tras varias reuniones con el tutor del proyecto ya se conocía la funcionalidad base que iba a tener la aplicación. Además, tras sucesivas reuniones y comunicación con dicho tutor, fue apareciendo funcionalidad nueva la cual era deseable para la aplicación y que además iba a tener un impacto en la fase de análisis y más concretamente también en el análisis de las interfaces de usuario.

Se llevaron a cabo más de una versión de prototipos, y para cada una de ellas se analizaron con profundidad todas las interfaces de ambos subsistemas, web y móvil.

1. **Primera versión** realizada tras la segunda reunión con el tutor del proyecto. En ella, tras plantear las principales funcionalidades del sistema se dio paso a esta primera versión.
2. **Segunda versión** realizada tras la tercera reunión con el tutor del proyecto. En ella se vieron carencias en la primera versión realizada ya que no cubrían todos los casos de uso que se fueron planteando durante la fase de análisis. Además, con el fin de mejorar la usabilidad de la aplicación se realizaron algunos cambios en algunas de las ya existentes.
3. **Tercera versión** que surge tras diversas pruebas con la segunda versión se decide: resituar algunos elementos, esto es, se detectan demasiadas complicaciones para interactuar con la vista debido al posicionamiento de determinados elementos. Para simplificar el uso de la aplicación se decide también, en algunos casos, la unificación de dos vistas en una sola. Se elimina funcionalidad que se consideró innecesaria y se añadió otra.

4.5.1 Primera versión de prototipos

Esta primera versión surge a raíz de los primeros requisitos funcionales para la aplicación. Suple la funcionalidad más básica y original de la misma.

4.5.1.1 Subsistema aplicación web

4.5.1.1.1 Login

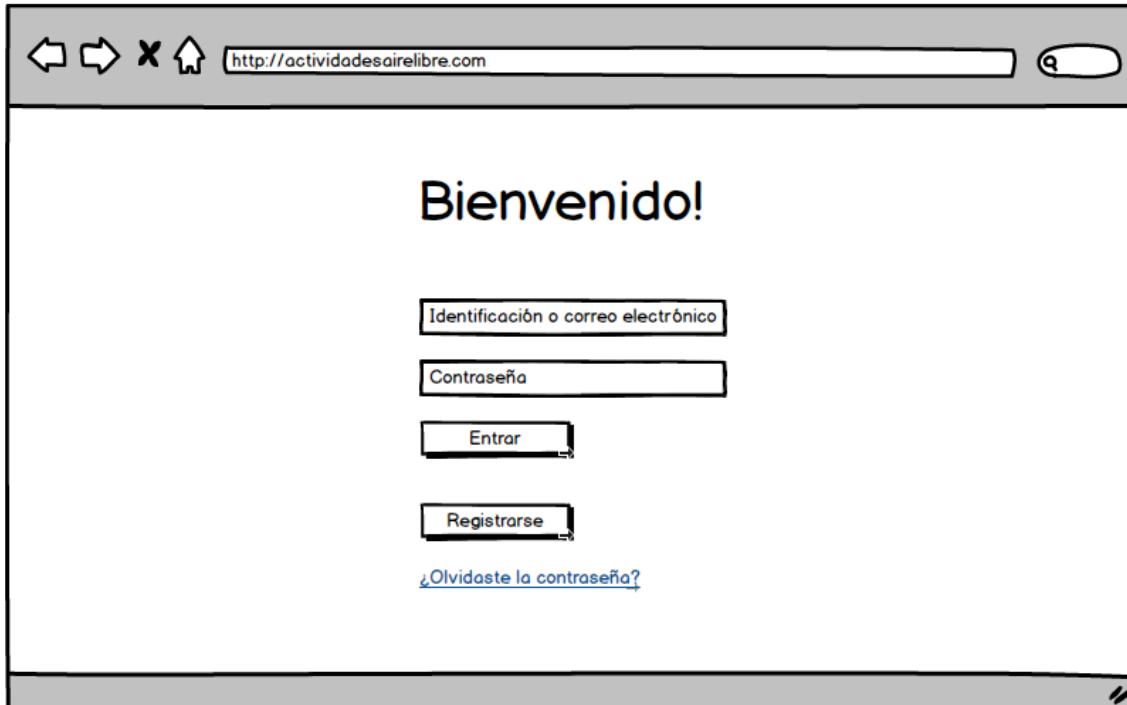


Figura 45 Prototipo de login - Primera versión

Desde la pantalla de login, la cual sería la inicial para cualquier conexión a la página web, se puede acceder tanto al listado de actividades como al formulario para recuperar la cuenta. Dicho formulario se realizaría a través de un pop-up al hacer click sobre el enlace.

4.5.1.1.2 Registro

Identificación

Correo electrónico

Contraseña

Repite contraseña

Registrarme

Tras registrarse, si los datos son correctos, se envía al usuario a la página de login y se muestra un popup de registro completado

Figura 46 Prototipo de registro - Primera versión

4.5.1.1.3 Página principal

Fecha de la actividad	Lugar de salida	Plazas máximas	
02-02-2017	---	10	Mas información
02-03-2017	---	14	Mas información
02-04-2017	---	11	Mas información
02-05-2017	---	22	Mas información
02-06-2017	---	8	Mas información
02-07-2017	---	9	Mas información

Figura 47 Prototipo página principal - Primera versión

Se puede entender la página principal como una de las vistas más importantes de la aplicación. Desde ella se puede acceder a casi cualquier otra a través de la barra de navegación situada en la parte superior de la ventana.

En ella se pueden encontrar y filtrar todas las actividades que aún no han sido realizadas, así como ir a cualquier otra vista de la aplicación.

Además, a partir de esta vista y para el resto de las de la aplicación, en todas aquellas restringidas a usuarios identificados, en la parte superior de la pantalla se incluye una barra de navegación. Por medio de ella se puede ir a cualquier otra parte de la aplicación, desde cualquier otra.

4.5.1.1.4 Datos personales

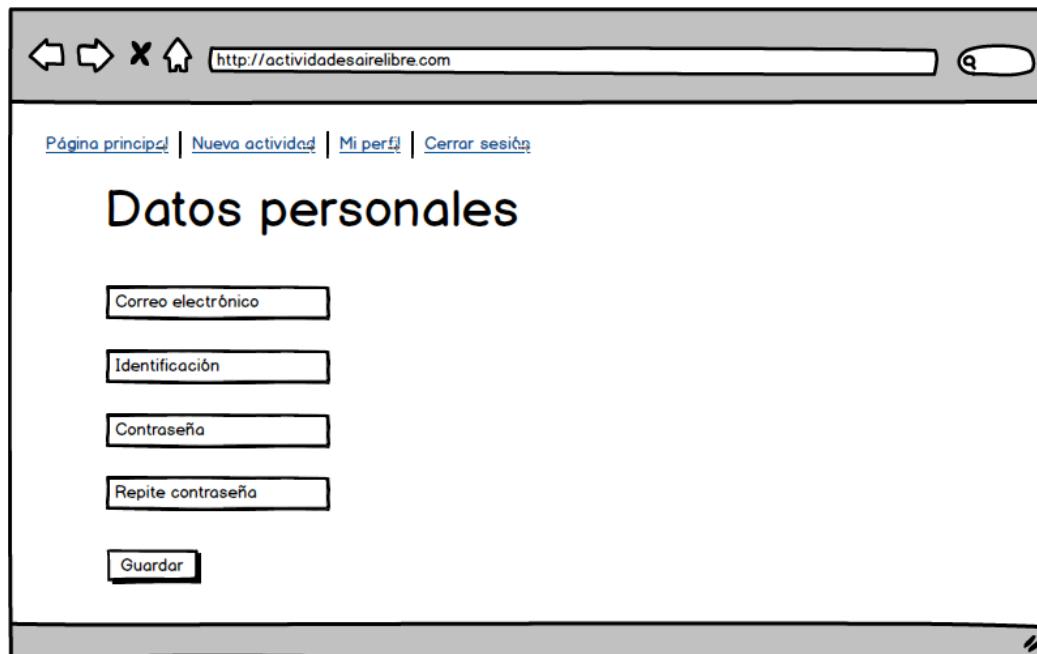


Figura 48 Prototipo Datos Personales - Primera versión

Cada campo de esta ventana sería editable y figuraría llenado con los datos personales del usuario que accediese a la vista. Cualquier usuario que esté identificado puede acceder a esta vista y desde ella ir a cualquier otra de la aplicación.

4.5.1.1.5 Nueva actividad

El prototipo muestra una interfaz web para la creación de una nueva actividad. La URL en la barra de dirección es <http://actividadesairelibre.com>. Los enlaces en la barra superior son 'Página principal', 'Nueva actividad', 'Mi perfil' y 'Cerrar sesión'. El título de la página es 'Nueva actividad'. Se requiere una 'Breve descripción de la actividad' (campo de texto). Una sección para 'Descripción detallada de la ruta, dificultad... o enlace donde se encuentre detallada MIDE' (campo grande que contiene el enlace 'MIDE'). Sección para 'Fecha y hora de comienzo de la actividad' (campo con calendario y hora). Sección para 'Plazas máximas' (campo numérico con un selector desplegable). Un enlace 'Especificificar track de la ruta' (enlace azul). Un botón 'Guardar actividad'.

Figura 49 Prototipo nueva actividad - Primera versión

Desde esta vista el cualquier usuario identificado puede crear una actividad nueva. Se incluye un formulario con todos los datos de la actividad.

En esta primera versión de los prototipos, el enlace “Especificificar track de la ruta” estaba pensado para abrir una ventana emergente con un mapa en el cual mediante una polilínea dibujar el supuesto track para la actividad.

El enlace “MIDE” se trata de un enlace a una página web externa donde opcionalmente poder realizar el detalle de la excursión, siendo también opcional especificarlo a modo texto en el cuadro de texto indicado para ello.

En sucesivas versiones de los prototipos, a raíz de cambios en los requisitos de la aplicación, veremos que tanto el campo MIDE como con el track surgen diferentes cambios.

4.5.1.1.6 Track de la ruta

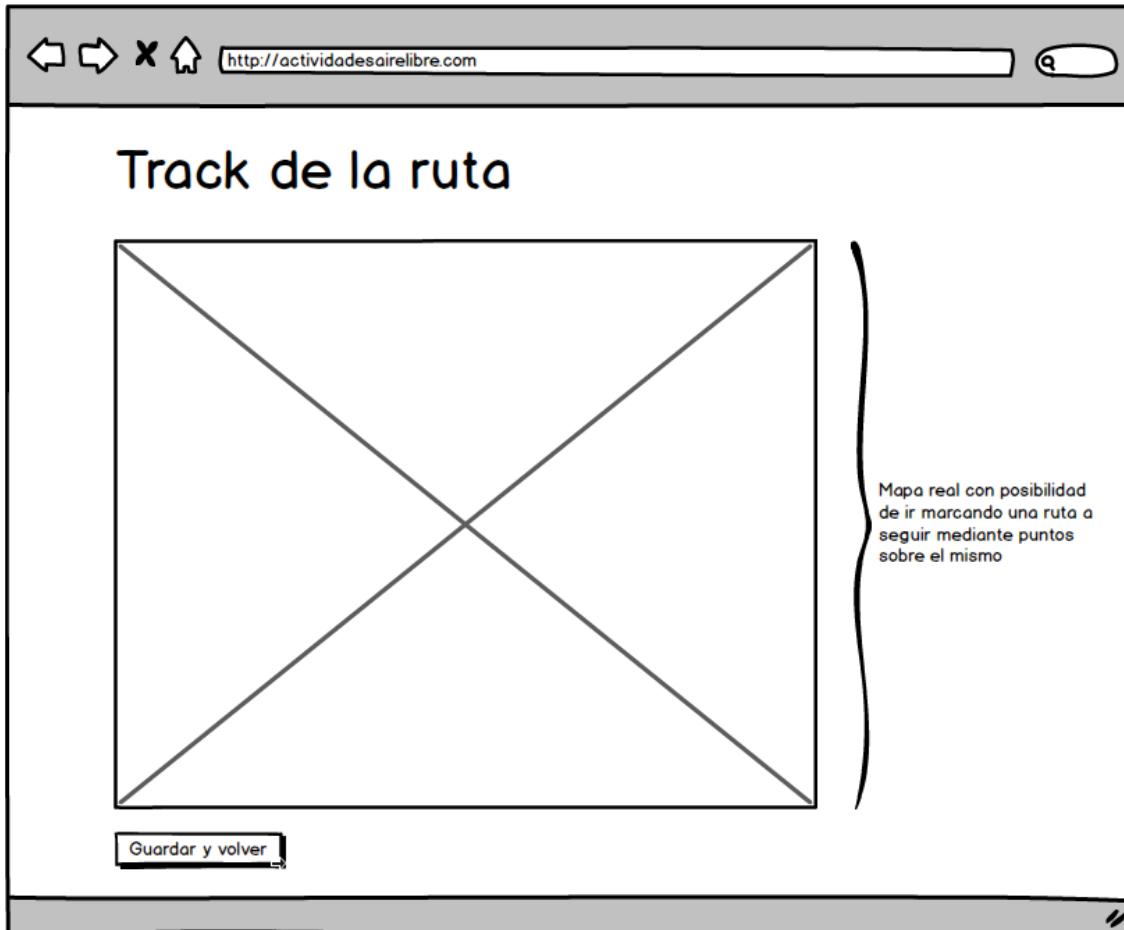


Figura 50 Prototipo track de la ruta - Primera versión

Se trata de la ventana emergente que se abriría al acceder al enlace de “Especificificar track de la ruta” en el prototipo de creación de actividad.

En esta vista solamente se mostraría un mapa al usuario con diferentes herramientas sobre el mismo para dibujar por medio de puntos conectados entre sí. El botón de “Guardar y volver” cerraría esta ventana y volvería a la vista de crear actividad.

4.5.1.1.7 Actividad



Figura 51 Prototipo vista de actividad - Primera versión

En esta vista se muestran todos los datos de una actividad, sea cual sea su estado, al usuario. Desde ella además se pueden acceder a otras vistas como el mapa de la ruta, la edición o monitorización de la actividad.

Tal y como se pensó el caso de uso tras la primera reunión con el tutor del proyecto, el enlace de mapa de ruta abriría una ventana emergente como la descrita en 4.5.1.1.6 con el mapa esta vez mostrando la ruta a seguir.

El resto de campos que se muestran en la vista al usuario no estarían disponibles para su edición.

4.5.1.1.8 Editar actividad

El prototipo muestra una interfaz web con el siguiente diseño:

- Barra superior:** Botones para navegar entre páginas y un campo de dirección con el valor "http://actividadesairelibre.com".
- Navegación:** Enlaces a "Página principal", "Nueva actividad", "Mi perfil" y "Cerrar sesión".
- Título:** "Editar actividad".
- Campo descriptivo:** "Breve descripción de la actividad" con el valor "Actividad de senderismo".
- Área descriptiva:** "Descripción detallada de la ruta, dificultad... o enlace donde se encuentre detallada MIDE" (que es lo que aparece en la captura).
- Fecha y hora:** "Fecha y hora de comienzo de la actividad" con un campo de tipo fecha y hora.
- Plazas máximas:** "Plazas máximas" con un campo desplegable que muestra el valor "14".
- Otros:** Un enlace "Cambiar track de la ruta" y un botón "Guardar cambios".

Figura 52 Prototipo editar actividad - Primera versión

En esta vista se muestran todos los campos que se mostraban en la vista de la actividad, pero en este caso editables. A ella podría acceder, tal y como se planteó para esta primera versión de los prototipos, el mismo usuario que creó la actividad.

El enlace a “Cambiar track de la ruta” funciona como se ha especificado en anteriores vistas, abriría una vista sobre la actual con un mapa y esta vez, editable, el track que se seguiría en la actividad.

4.5.1.1.9 Monitorizar actividad

Este prototipo es una captura de pantalla de un navegador web. La URL es <http://actividadesairelibre.com>. La barra superior tiene botones para retroceder, avanzar, cerrar y un icono de usuario. Los enlaces en la barra superior son: Página principal, Nueva actividad, Mi perfil y Cerrar sesión.

Monitorización actividad

Personas con monitorización activa

Identificación	Último reporte de posición a las	
Usuario 1	10:10 am	Ver última posición sobre mapa
Usuario 2	10:10 am	Ver última posición sobre mapa
Usuario 3	10:10 am	Ver última posición sobre mapa
Usuario 4	10:10 am	Ver última posición sobre mapa
Usuario 5	10:10 am	Ver última posición sobre mapa
Usuario 6	10:10 am	Ver última posición sobre mapa

[Ver progreso de la actividad](#)

Los enlaces a la posición del mapa abren una nueva ventana sobre la existente con un mapa y la posición del último reporte de la persona sobre la que se ha seleccionado ver la información

Abrirá una nueva ventana sobre la existente con un mapa con la información del track de la ruta y la posición sobre ella de los participantes en la actividad

Figura 53 Prototipo monitorizar actividad - Primera versión

Esta vista es una de las más significativas de la aplicación web, ya que en ella se muestra toda la información relativa a la situación de los excursionistas en una actividad en curso.

Para la primera versión de los prototipos se planteó que a ella solo pudiese acceder el mismo usuario que creó la actividad. Sería accesible solamente desde la página del detalle de la actividad.

En el centro de la vista, en una tabla scrollable aparecerían listados los participantes de la actividad que estaban siendo monitorizados. Para cada participante se mostraría tanto la hora del último reporte de posición conocido por la actividad como un enlace para verlo sobre el mapa. En sucesivas versiones, como se verá más adelante, esta tabla se decide paginarla ante la posibilidad de aparecer múltiples elementos en el listado.

De la misma forma que funcionaba en vistas anteriores, el enlace “Ver última posición sobre mapa” fue pensado para abrir una ventana emergente con la vista de un mapa y sobre éste señalada la posición del excursionista.

Tal y como se verá en la siguiente versión de los prototipos de la aplicación, este prototipo apenas cubría los casos de uso planteados para la monitorización de una actividad ya que no da lugar a comparar la posición de un excursionista con la del resto y con la del track de la ruta.

4.5.1.2 Subsistema aplicación móvil

4.5.1.2.1 Acceso excursionista



Figura 54 Prototipo acceso excursionista - Primera versión

Esta sería la vista de acceso que se mostraría a un usuario que no tuviera una cuenta de usuario vinculada con el dispositivo.

En la primera versión de los prototipos se había pensado el acceso a la aplicación de dos maneras, por medio de un login para aquellos registrados en la página web (guías de actividades) y un acceso simplemente aportando nombre y apellidos para cualquier otro usuario que fuese a ser monitorizado en una actividad. Como se verá más adelante, esta segunda funcionalidad quedaría desechada.

4.5.1.2.2 Acceso a guía



Figura 55 Prototipo acceso guía - Primera versión

La diferencia entre esta vista y la anterior es que esta sí es un logín de usuario en el que se tienen que aportar las credenciales que se utilizaron durante el registro en la página web.

A través de esta vista se accedería a la aplicación con rol de “Guía”. Más adelante se unificaría todo bajo el mismo acceso debido a que el rol de guía estaría diferenciado en función de cada actividad.

4.5.1.2.3 Actividades

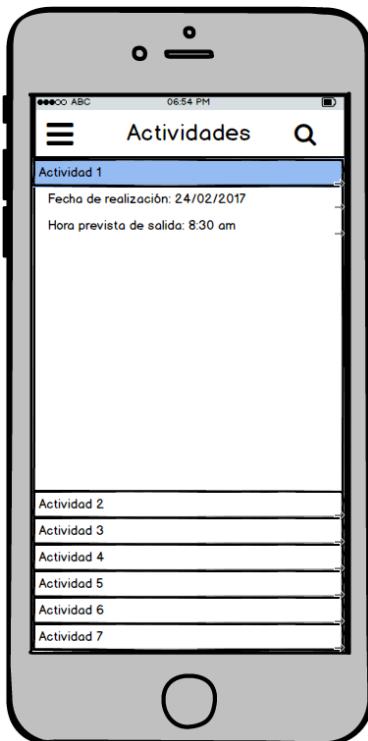


Figura 56 Prototipo Actividades – Primera versión

En esta vista, para cualquier tipo de usuario identificado en la aplicación se muestra la lista de actividades en el sistema que aún no han sido realizadas o están en curso en ese momento. Además del nombre de cada actividad se muestra, para cada una de ellas, la fecha de realización y la hora.

Mediante la barra de herramientas se puede acceder al menú superior izquierda (Navigation Drawer) desde el cual navegar entre vistas de la aplicación.

Mediante el ícono de búsqueda situado en la parte superior derecha se podrían filtrar las actividades por fecha de realización, a través de un calendario emergente.

Haciendo click en cualquiera de las actividades listadas en esta vista se accedería a la vista particular de la actividad.

4.5.1.2.4 Actividad

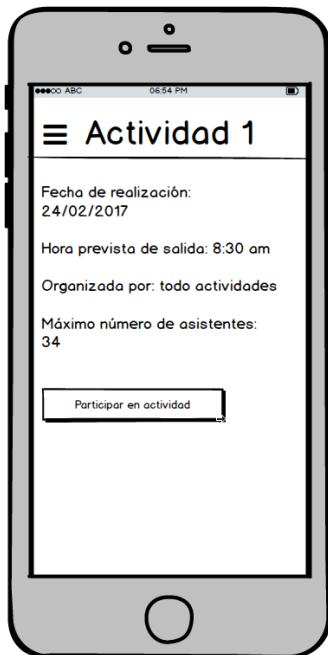


Figura 57 Prototipo actividad - Primera versión

En esta vista se muestra al usuario toda la información para una actividad que esté en curso o pendiente de realización. Desde la misma, si la actividad está en curso, se puede acceder a la participación. A través de ese enlace el usuario pasaría a ser monitorizado por el sistema.

4.5.1.2.5 Monitorización



Figura 58 Prototipo monitorización – Primera versión

Esta vista es accesible desde el menú presente en todas las vistas accesible desde la parte superior izquierda de la aplicación.

En ella el guía de la actividad puede acceder a la posición de todas las personas que están siendo monitorizadas en la actividad.

Tal y como pasaba en su vista homóloga en la aplicación web, y como se verá en la segunda versión de los prototipos, esta vista carece de mucha de la funcionalidad que se planteó durante la fase de análisis.

En este caso el enlace “Ver posición excursionistas” abriría una actividad emergente sobre la actual con un mapa y la posición marcada sobre éste de todos los asistentes a la actividad.

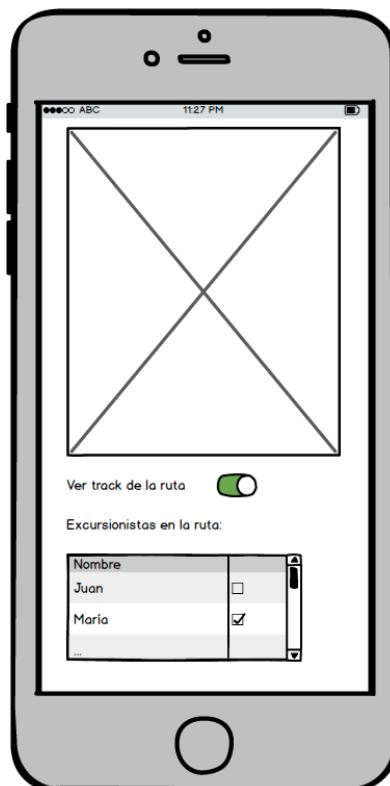


Figura 59 Prototipo monitorización excursionistas – primera versión

4.5.2 Segunda versión de prototipos

Esta versión, como se describía anteriormente, se planteó tras una tercera reunión con el tutor del proyecto tras descubrir, a raíz de la primera, que había casos de uso no cubiertos de manera correcta. Además, se añadieron nuevas funcionalidades a la aplicación y se modificaron otras, por lo que fue necesario la realización de esta segunda versión.

Se analizarán a continuación todas aquellas vistas que por estos motivos hayan cambiado, dándose a entender por tanto que las que no se mencionen ahora será porque se mantienen igual que en la primera versión.

4.5.2.1 Subsistema aplicación web

4.5.2.1.1 Página principal

Outdoor Collective Activities Monitorization

<http://ocam.com>

Los filtros "Solo mis actividades" y "Solo actividades iniciadas" solo serán visibles para aquellas personas identificadas

Página principal | Nueva actividad | Mi perfil | Cerrar sesión

Actividades pendientes de realización

Solo mis actividades Solo actividades iniciadas

Fecha de la actividad	Lugar de salida	Plazas máximas	
02-02-2017	---	10	Mas información Monitorizar
02-03-2017	---	14	Mas información Monitorizar
02-04-2017	---	11	Mas información Monitorizar
02-05-2017	---	22	Mas información Monitorizar
02-06-2017	---	8	Mas información Monitorizar
02-07-2017	---	9	Mas información Monitorizar

Figura 60 Prototipo (v2) página principal

Se decide además para esta versión de los prototipos en la página principal que se añada a la cabecera de cada columna de la tabla la ordenación de las actividades según dicha cabecera.

4.5.2.1.2 Nueva actividad

El prototipo muestra la siguiente interfaz:

- Breve descripción de la actividad:** Un campo de texto.
- Descripción detallada de la ruta, dificultad... o enlace donde se encuentre detallada:** Un campo de texto con el placeholder "MIDE".
- Fecha y hora de comienzo de la actividad:** Campos para seleccionar fecha y hora.
- Plazas máximas:** Un campo numérico establecido en 10.
- Cargar fichero (.gpx) de track de la ruta:** Un botón para cargar un archivo.
- Especificar guía/s:** Un botón que abre una tabla emergente.
- Tabla de guías:** Una tabla que muestra los siguientes datos:

Nombre	Correo electrónico	Opciones
Víctor Castaño	victor@gmail.com - Guía	<input checked="" type="checkbox"/>
Juan García	juan@gmail.com - Guía	<input type="checkbox"/>
...		
- Guardar actividad:** Un botón para guardar la actividad.

Una nota en la parte derecha explica la tabla de guías:

Esta tabla no tendrá contenido hasta no especificarse los guías (botón superior). Una vez especificados figurarán ahí. Mediante el checkbox, al desmarcar uno de ellos se borrará automáticamente de la tabla. En este caso Víctor Castaño es el usuario logueado en la aplicación, y como tal no puede eliminarse de ser guía

Figura 61 Prototipo (v2) nueva actividad

Esta vista sufre diversos cambios con respecto a la anterior. De primeras se decide suprimir el “editor” de track de la ruta y reemplazarlo por un archivo con extensión “.gpx” que será subido a través de un campo input en la propia vista.

Se añade una selección de guías para la actividad. Dicha selección se realizaría a través de una ventana emergente “Añadir guías” y los cambios se verían reflejados en la tabla inmediatamente inferior a dicho botón.

Este prototipo es una interfaz web para la selección de guías. En la parte superior hay dos campos de texto para filtrar por nombre y por correo electrónico. Abajo se muestra una lista de guías con columnas para Nombre, correo electrónico y añadido (que incluye un checkbox). Los checkboxes están marcados para Juan Martínez y Laura García. A continuación, hay un botón 'Aplicar cambios'.

Nombre	correo electrónico	añadido
Víctor Castaño	victor@gmail.com	<input type="checkbox"/>
Juan Martínez	juan@gmail.com	<input checked="" type="checkbox"/>
Laura García	laura@gmail.com	<input type="checkbox"/>
...		

Figura 62 Prototipo selección de guías.

Posteriormente la edición y la visualización de la actividad vendrían dadas por el mismo cambio con los guías de la ruta acontecido con la creación de la actividad y dispuesto en la ventana de la misma manera.

4.5.2.1.3 Monitorización

Este prototipo es una interfaz web para la monitorización de actividades. La barra superior muestra el título 'Outdoor Colective Activities Monitorization' y el URL 'http://ocam.com'. La barra inferior tiene enlaces a 'Página principal', 'Nueva actividad', 'Mi perfil' y 'Cerrar sesión'.

Monitorización actividad

Personas con monitorización activa

Identificación	Último reporte de posición a las	
Usuario 1	10:10 am	Ver última posición sobre mapa
Usuario 2	10:10 am	Ver última posición sobre mapa
Usuario 3	10:10 am	Ver última posición sobre mapa
Usuario 4	10:10 am	Ver última posición sobre mapa
Usuario 5	10:10 am	Ver última posición sobre mapa
Usuario 6	10:10 am	Ver última posición sobre mapa

Los enlaces a la posición del mapa abren una nueva ventana sobre la existente con un mapa y la posición del último reporte de la persona sobre la que se ha seleccionado ver la información.

[Ver progreso de la actividad](#) → Abrirá una nueva ventana sobre la existente con un mapa con la información del track de la ruta y la posición sobre ella de los participantes en la actividad.

[Mandar mensaje a guías](#) → Botón solo visible para los guías de la actividad. Abrirá una ventana emergente sobre la actual donde se escribirá el mensaje.

Figura 63 Prototipo (v2) monitorización actividad

Con respecto al prototipo anterior, se añade el acceso a mandar un mensaje a los guías, que tal y como viene explicado en el propio prototipo, se realizaría a través de una ventana emergente.

Finalmente, otra diferencia de esta ventana con la anterior, y con el fin de cubrir nuevos requisitos, es la ventana de monitorización. En la versión anterior no se podía ver la

actividad sobre el mapa de un excursionista. Para suplir esa carencia se decidió trabajar más la vista de monitorización con el fin de poder observar a varios excursionistas, fuesen guías o no, a la vez.

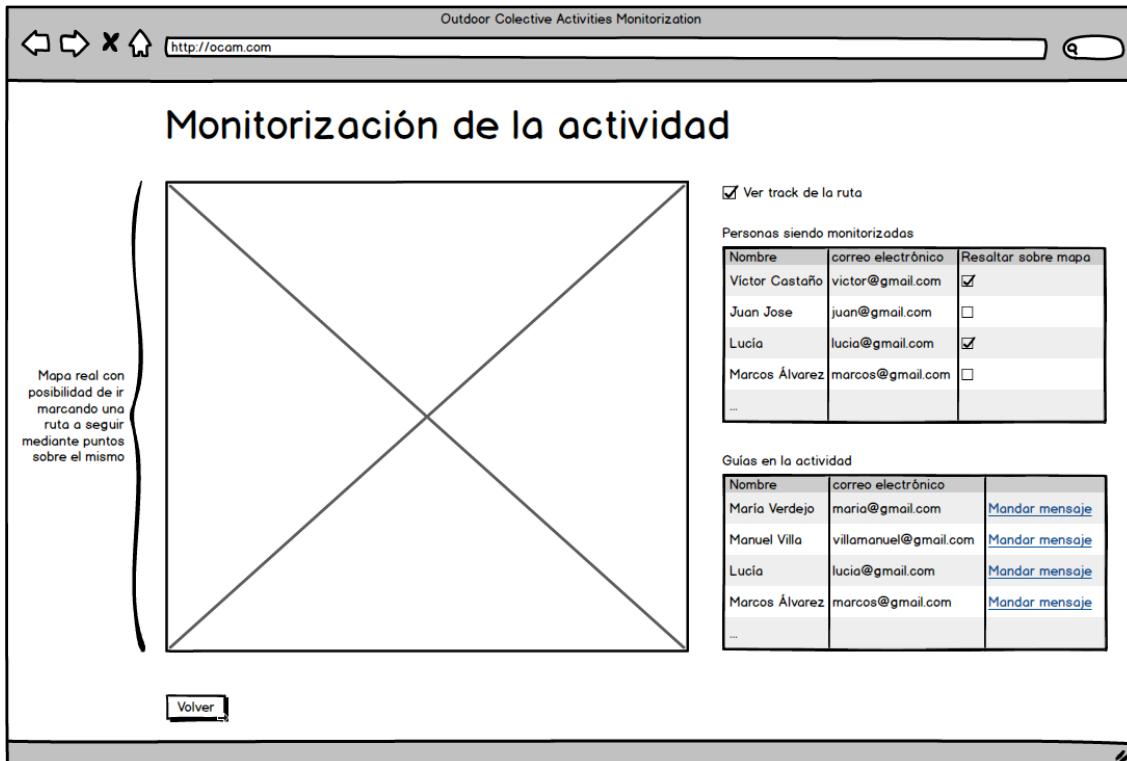


Figura 64 Prototipo (v2) monitorización sobre mapa

4.5.2.2 Subsistema aplicación móvil

4.5.2.2.1 Vinculación de la cuenta

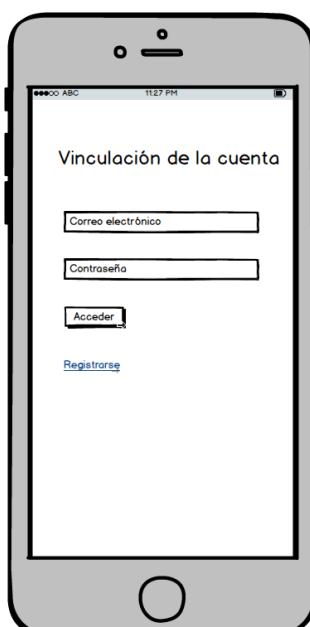


Figura 65 Prototipo (v2) Vinculación de la cuenta

Con el fin de unificar la funcionalidad con respecto a el acceso de guías y el “acceso anónimo”, se decide unificar todo bajo una misma vinculación. En este caso el requisito funcional había cambiado también, añadiendo ahora que un usuario para acceder a la aplicación web, deba también estar registrado previamente.

4.5.2.2.2 Actividad

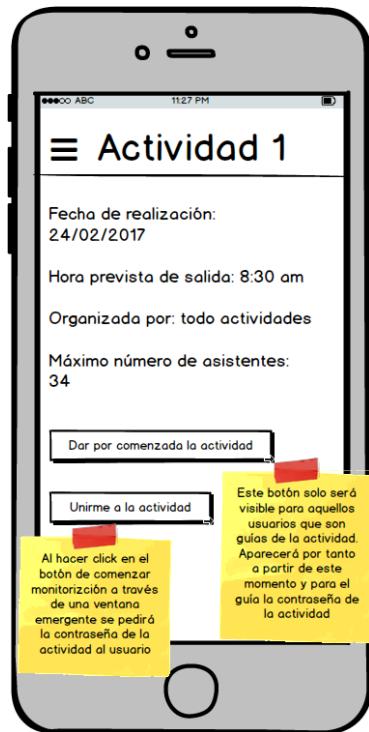


Figura 66 Prototipo (v2) Actividad

Se añade la funcionalidad, también por medio de un botón, con respecto a la versión anterior, para unirse a una actividad.

4.5.3 Tercera versión de prototipos

Surge entre otras cosas y como se comentó anteriormente, tras problemas detectados con el uso de la aplicación, en su mayoría de usabilidad. Se eliminan vistas por unificación con otras y elementos en las mismas por desestimación de diferentes funcionalidades.

De la misma manera que con la segunda versión, solamente se analizarán aquellas interfaces que hayan sufrido cambios con respecto a la segunda versión. Aquellas que no figuren en esta se entenderá que permanecen igual.

4.5.3.1 Subsistema aplicación web

4.5.3.1.1 Acceso

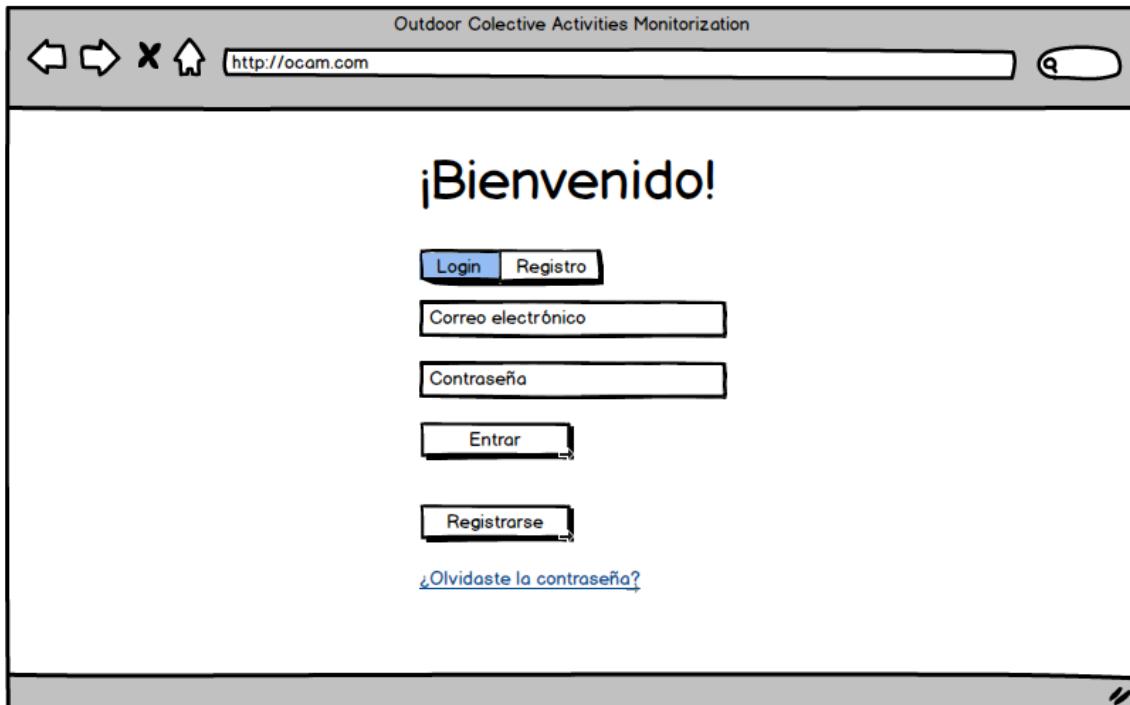


Figura 67 Prototipo (v3) login

En este caso el login y el registro se decide unificarlo en una misma vista. De cualquier manera, para no tener presentes en pantalla al mismo tiempo ambos formularios, los cuales requieren de campos diferentes cada uno de ellos, se decide hacerlo a través de tabs.

Haciendo click sobre "Registro" aparecería el mismo formulario del registro pudiendo volver al de login a través de "Login".

4.5.3.1.2 Actividades

Los filtros "Solo mis actividades" y "Solo actividades iniciadas" solo serán visibles para aquellas personas identificadas

Página principal | Nueva actividad | Mi perfil | Cerrar sesión

Actividades pendientes de realización

Actividades pendientes/Realizadas ▾

Fecha de la actividad	Plazas	Descripción	estado
02-02-2017	10	Descrp.	Pendiente / En curso
02-03-2017	14	Descrp.	Pendiente / En curso
02-04-2017	11	Descrp.	Pendiente / En curso
02-05-2017	22	Descrp.	Pendiente / En curso
02-06-2017	8	Descrp.	Pendiente / En curso
02-07-2017	9	Descrp.	Pendiente / En curso

En el caso de la descripción, estaría el texto truncado en caso de superar el ancho de la columna

Figura 68 Prototipo (v3) actividades

Se decide realizar a través de un combobox y no un checkbox el filtro de actividades pendientes y actividades realizadas. De esta forma no se permite al usuario ver las actividades de un tipo a la vez que las del otro.

Además, se decide por cuestiones de requisitos cambiar el contenido de la tabla. Haciendo click sobre la cabecera de cada columna se realizaría una ordenación ascendente/descendiente de los elementos en dicha columna.

4.5.3.1.3 Nueva actividad

Página principal | Nueva actividad | Mi perfil | Cerrar sesión

Nueva actividad / Editar / Ver Guías

Breve descripción de la actividad

Descripción detallada de la ruta, dificultad... o enlace donde se encuentre detallada

Enlace a descripción

Plazas máximas : 10

Fecha y hora de comienzo de la actividad / /

Cargar fichero (.gpx) de track de la ruta

Añadir Suprimir todos

Nombre	
Nombre	Eliminar

La vista sería la misma tanto para creación, edición o visualización. Cambiaría la posibilidad de editar los campos o no.
El botón de Guardar solo estaría disponible en creación y edición, mientras que el de monitorización solo figuraría para visualización.

Monitorizar actividad

Guardar actividad

Figura 69 Prototipo (v3) nueva actividad

El contenido de la pantalla es muy similar al de la segunda versión del prototipo. El único cambio es el de cómo añadir los guías. En la segunda versión de los prototipos se realizaba a través de una ventana auxiliar. Se ha decidido por cuestiones de usabilidad integrarlo todo en la misma.

A través del botón “añadir” se mostraría un pop-up para escribir el nombre del guía, a diferencia de la segunda versión que se seleccionaba de un listado. Esto podría convertirse en un problema a la larga cuando los elementos en dicho listado creciesen demasiado.

Por otro lado, se ha decidido realizar una reorganización de los elementos de la pantalla aprovechando el ancho de la misma. Sin embargo, para pantallas con menores resoluciones se mantendría el esquema de la segunda versión, mostrándose todo en la misma columna.

Tal y como figura en una anotación en el prototipo, esta vista serviría tanto para ver una actividad, como para crearla como para editarla. Lo único que cambiaría sería el botón de confirmar cambios y la posibilidad de edición de los campos.

4.5.3.1.4 Monitorización

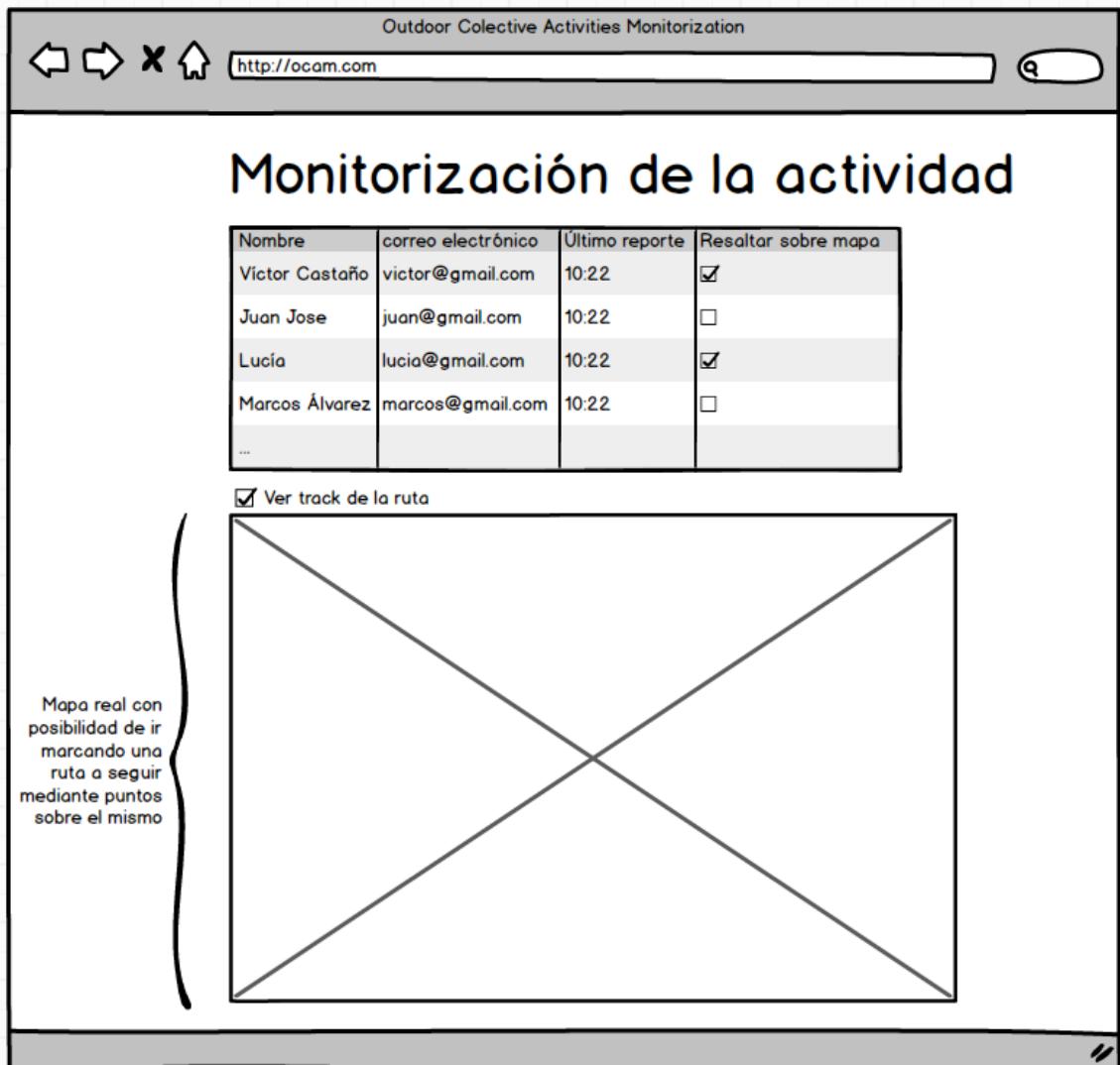


Figura 70 Prototipo (v3) monitorización

De la misma manera que sucede con el prototipo de crear una actividad, por motivos de mejorar la usabilidad de esta ventana se decide unificarlo todo en la misma, a diferencia de la segunda versión en la cual esta funcionalidad estaba partida en dos ventanas diferentes.

La información de los participantes con el acceso a mostrar la información sobre el mapa ahora está situada encima del propio mapa, lo cual hace que sea mucho más sencilla su manipulación.

Con el fin de no mezclar el scroll de la ventana con el del mapa se decide añadir la posibilidad de poder colapsar la tabla con la información de los participantes.

4.5.3.2 Subsistema aplicación móvil

4.5.3.2.1 Registro



Figura 71 Prototipo (v3) registro

El registro no sufre a penas cambios. Tras una redefinición de los requisitos para registrarse en la aplicación se suprimen los campos “Nombre” y “Apellidos” y se añade “login”.

4.5.3.2.2 Actividades

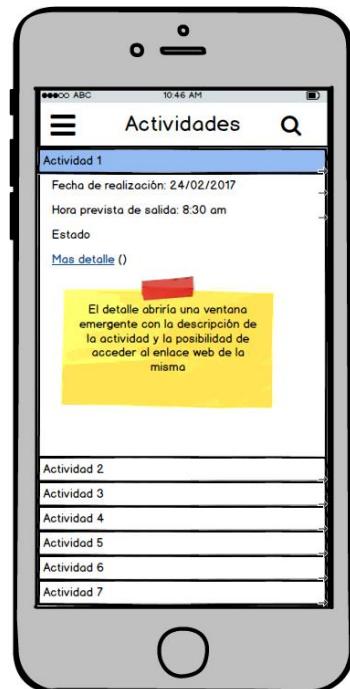


Figura 72 Prototipo (v3) Actividades

Se decide dar una mayor utilización al listado de las actividades aumentando la información que se muestra de cada una de ellas. Se añade para esto un enlace a abrir una ventana pop-up con la descripción de la actividad y un acceso directo al enlace MIDE de la misma.

4.5.3.2.3 Actividad

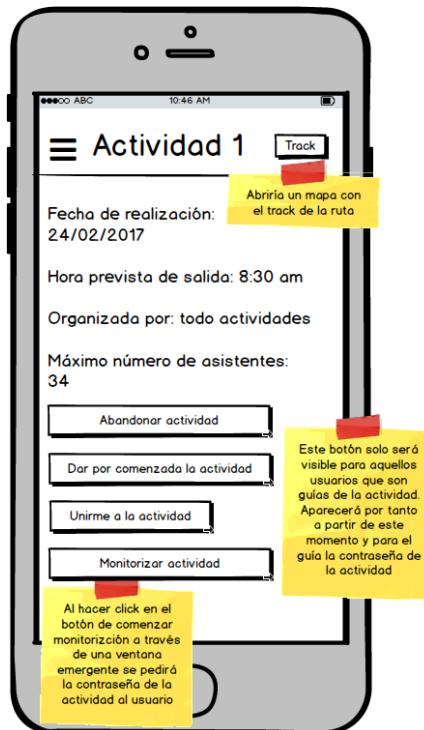


Figura 73 Prototipo (v3) actividad

De la misma manera que con el prototipo de la lista de actividades, en esta vista se decide concentrar toda la funcionalidad relacionada con la actividad. Para ello se deciden añadir accesos a la gestión de la misma por medio de botones: abandonar, comenzar, unirse y monitorizar.

Todos ellos no estarán nunca visibles a la vez para el usuario, dependiendo del estado de la actividad y de la relación usuario-actividad.

4.5.3.2.4 Monitorización

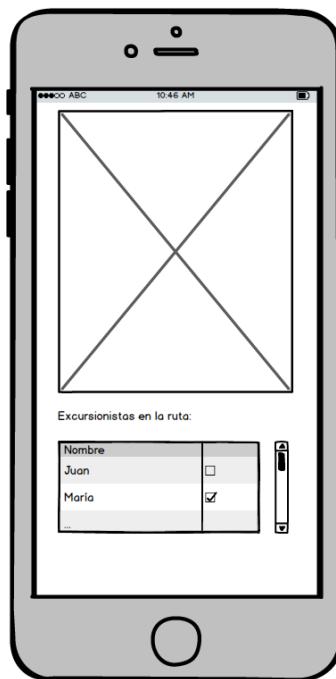


Figura 74 Prototipo (v3) monitorización

De la misma manera que pasó con su análogo en la página web, la diferencia con la segunda versión de los prototipos es que por motivos de aumentar la facilidad de uso se decide agrupar dos pantallas en la misma.

En este caso la información de la actividad no se veía de relevancia situarla en esta ventana pues ya está presente en la de la propia actividad. La información de los participantes se decide situarla debajo del mapa para tener un mejor acceso a la misma.

4.5.3.2.5 Perfil



Figura 75 Prototipo perfil

Para la tercera versión de los prototipos se observa una carencia de funcionalidad de la aplicación móvil, la cual no permitía el cambio de contraseña.

Para ello se decide introducir una nueva pantalla de perfil, muy similar a la existente en la aplicación web donde el usuario puede cambiar su contraseña.

4.5.4 Mapa de navegación definitivo

4.5.4.1 Subsistema aplicación web

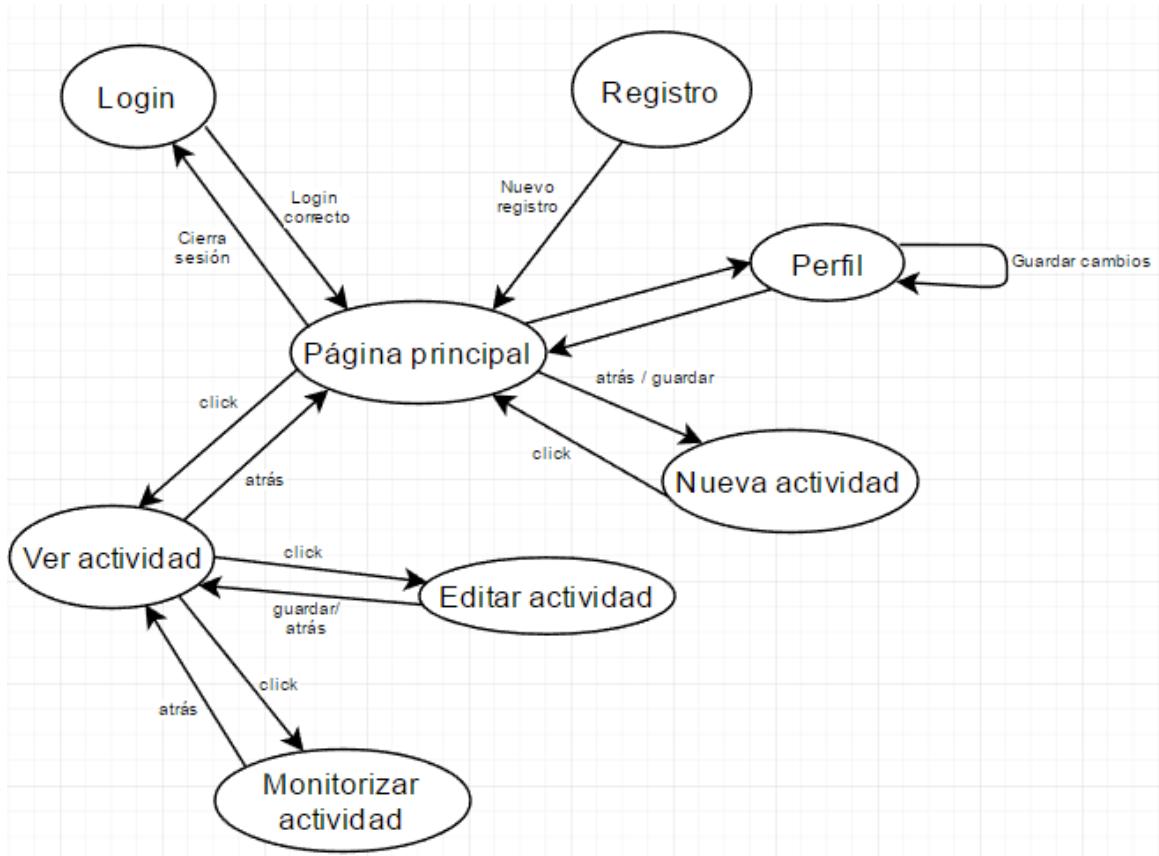


Figura 76 Mapa de navegación – subsistema aplicación web

El punto de partida de cualquier usuario en la aplicación web será la página principal. Desde ahí podrá acceder a cualquier otra a través de la barra de navegación.

4.5.4.2 Subsistema aplicación móvil

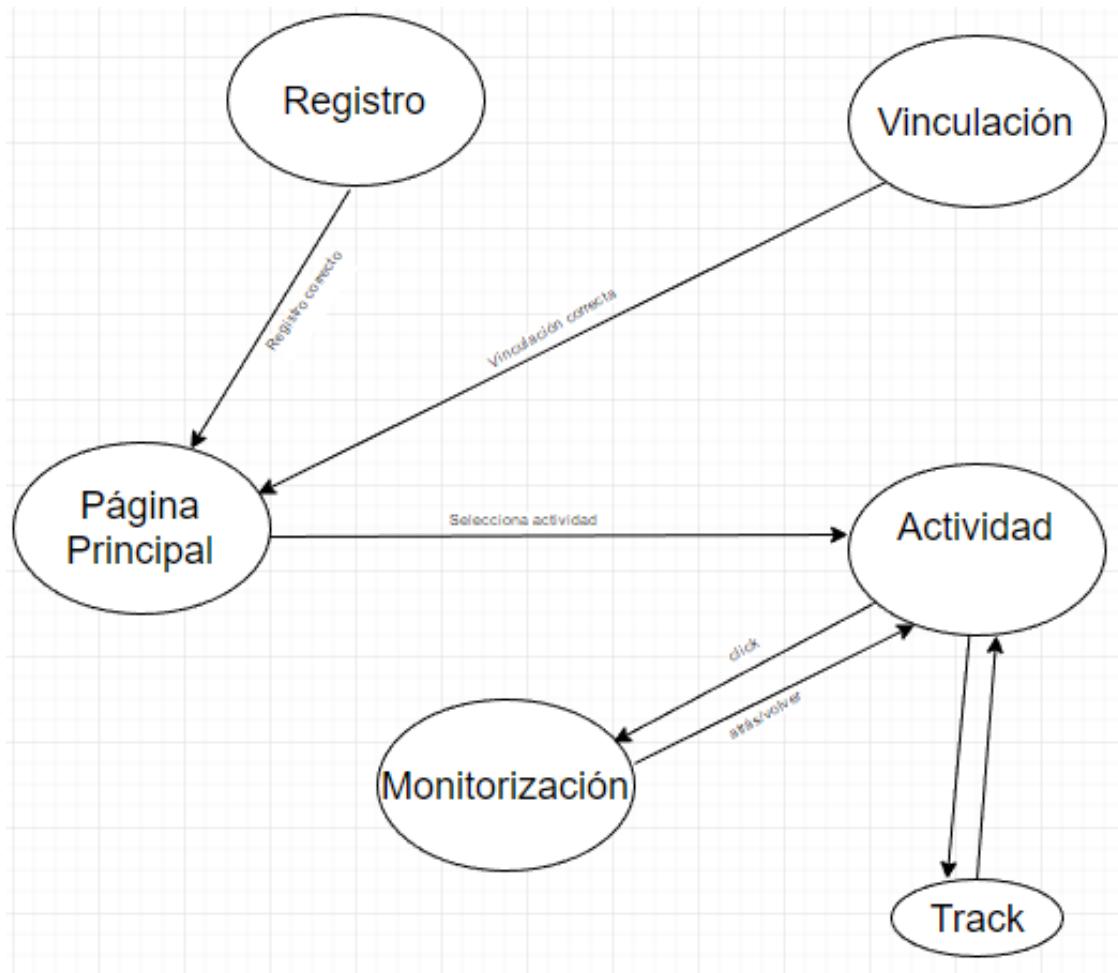


Figura 77 Mapa de navegación – aplicación móvil

En este caso, cualquier usuario entraría por la vista de vinculación, salvo que tenga ya la vista vinculada en cuyo caso pasaría a la página principal. Así mismo, en el mapa de navegación se puede observar que la principal diferencia con la segunda versión de los prototipos es que se reduce el número de vistas en la aplicación móvil debido a haber unificado varias en la misma.

4.5.5 Matriz de trazabilidad interfaces - casos de uso subsistema web

	CU-W-1 Registrarse	CU-W-2 Iniciar sesión	CU-W-3 Ver actividades	CU-W-4 Recuperar cuenta	CU-W-5 Cerrar sesión	CU-W-6 Borrar cuenta	CU-W-7 Crear actividad	CU-W-8 Editar actividad	CU-W-9 Cambiar datos personales	CU-W-10 Monitorizar actividad	CU-W-11 Eliminar actividad
Acceso	✓	✓		✓							
Actividades			✓		✓						
Actividad					✓		✓	✓			✓
Monitorización					✓					✓	
Datos personales					✓	✓			✓		

Tabla 14 Matriz de trazabilidad interfaces con casos de uso subsistema web

4.5.6 Matriz de trazabilidad interfaces – casos de uso subsistema aplicación móvil

	CU-M-1 Registrarse	CU-M-2 Vincular cuenta	CU-M-3 Ver actividades	CU-M-4 Comenzar actividad	CU-M-5 Monitorizar actividad	CU-M-6 Unirse actividad	CU-M-7 Finalizar actividad	CU-M-8 Cambiar password	CU-M-9 Abandonar actividad	CU-M-10 Cerrar sesión
Login		✓								
Registro	✓									
Actividades			✓							✓
Actividad				✓		✓	✓			
Monitorización					✓					
Perfil								✓		✓

Tabla 15 Matriz de trazabilidad interfaces con casos de uso subsistema aplicación móvil

4.6 Especificación del Plan de Pruebas

En esta sección se especifica el plan de pruebas previo al diseño y la implementación del sistema. Se va a dividir la especificación según los dos subsistemas aplicación web y aplicación móvil, y para cada caso de uso de cada uno de ellos. Teniendo presentes todos los posibles caminos de cada diagrama de actividad y el flujo de acciones que causan dichos caminos alternativos en cada caso de uso, se va a tratar de cubrir todos ellos mediante los casos de prueba.

Además, se van a realizar las pruebas no solo en las condiciones óptimas del sistema, para así poder detectar todos los defectos posibles derivados de ello. Para cada caso de prueba se distingue:

- Identificador: identificador único para cada caso de prueba.
- Descripción: una descripción breve del caso de prueba.
- Procedimiento: conjunto de pasos a seguir para reproducir el caso de prueba.
- Salida esperada: la respuesta del sistema esperada a la ejecución del caso de prueba.

4.6.1 Subsistema página web

4.6.1.1 CU-W-1 Registrarse

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 1	Registro correcto de un usuario	Se introduce en el campo correo electrónico correo@gmail.com (cuenta no registrada en el sistema) y como contraseña 2222\$2222 y en repetir contraseña.	En el sistema se habrá registrado con éxito la cuenta de usuario correo@gmail.com con password 2222\$2222
Caso de prueba 2	Registro inválido por contraseña	Se introduce en el campo correo electrónico el correo correo@gmail.com (cuenta no registrada en el sistema) y en la password 2222\$2222. En el campo repite password introducimos 2221\$2222.	El sistema no debería registrar la nueva cuenta de usuario y debería notificar de que las contraseñas no coinciden.
Caso de prueba 3	Registro inválido por correo electrónico	Se introduce en el campo correo electrónico el correo correo1@gmail.com (cuenta ya registrada en el sistema) y en la password 2222\$2222. En el campo repite password introducimos 2222\$2222.	El sistema no debería registrar la nueva cuenta de usuario y debería notificar de que el correo electrónico utilizado ya está registrado en el sistema.
Caso de prueba 4	Registro inválido por correo datos incompletos	Se deja en blanco el campo de correo electrónico y en la password 2222\$2222. En el campo repite password introducimos 2222\$2222.	El sistema no debería proceder con el registro de la cuenta y avisaría al usuario con el campo requerido que está en blanco.

4.6.1.2 CU-W-2 Iniciar sesión

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 5	Iniciar sesión con contraseña incorrecta	Se introduce el correo electrónico válido y contraseña incorrecta.	El sistema no debería permitir el inicio de sesión.

4.6.1.3 CU-W-3 Ver actividades

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 6	Ver actividades como usuario invitado	Se accede al sistema como usuario invitado y se abre la vista de actividades.	El sistema mostrará las actividades registradas en el sistema, ordenadas decrecientemente por la fecha prevista de realización de la actividad.
Caso de prueba 7	Ver actividades realizadas como usuario invitado	Se intenta acceder al sistema a la vista de actividades realizadas.	El sistema no debería mostrar ninguna actividad
Caso de prueba 8	Idioma y ordenación de campos	Se accede al sistema como usuario identificado a través del login. Se accede a la lista de actividades y se cambia el idioma a inglés. Se reordenan las actividades por estado, primero descendiente y luego ascendientemente.	La ordenación, tanto descendiente como ascendiente debería realizarse con normalidad y el idioma debería seguir siendo el inglés.

4.6.1.4 CU-W-4 Recuperar cuenta

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 9	Recuperar cuenta no validada	Acceder al registro de la aplicación. Registrar una cuenta con correo electrónico 'testing@gmail.com', 'testing', '123', '123' valores respectivamente al formulario. Una vez registrada, sin validar la cuenta recuperar la contraseña utilizando el correo electrónico. Intentar acceder con la nueva contraseña.	Debería denegar el acceso a la aplicación puesto que la cuenta aún no ha sido validada.
Caso de prueba 10	Recuperar cuenta validada	Acceder al login y recuperar una cuenta que esté validada. Intentar acceder a la aplicación con la contraseña actual.	El sistema debería bloquear el inicio de sesión advirtiendo que las credenciales son inválidas.

4.6.1.5 CU-W-5 Cerrar sesión

Identificador	Descripción	Procedimiento	Salida esperada
130	Víctor Castaño Gutiérrez	Escuela de Ingeniería Informática - Universidad de Oviedo	

Caso de prueba 11	Cierre de sesión	Se accede al sistema como usuario registrado y se cierra sesión. Se intenta acceder a la vista de crear actividad	El sistema debería redireccionar a la pantalla de login y no reconocer el usuario logueado.
-------------------	------------------	---	---

4.6.1.6 CU-W-6 Borrar cuenta

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 12	Borrado cuenta	Se accede con una cuenta válida a la aplicación y se elimina a través del perfil. Se intenta acceder con las credenciales de la cuenta.	El sistema debería advertir de que las credenciales son inválidas.
Caso de prueba 13	Reutilización de datos de usuarios borrados	Se accede con una cuenta válida a la aplicación y se elimina a través del perfil. Intentar registrar una cuenta con un correo electrónico no utilizado y el login de la cuenta borrada.	El sistema debería bloquear el registro y advertir de que el nombre de usuario está en uso.

4.6.1.7 CU-W-7 Crear actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 14	Crear actividad con campos sin llenar	Se accede al sistema como usuario registrado y se abre la vista de crear actividad. Se deja sin llenar algún campo obligatorio.	El sistema no debería registrar la nueva actividad y avisaría al usuario del campo que debe ser cumplimentado.
Caso de prueba 15	Crear actividad con un track de ruta vacío	Se accede al sistema como usuario registrado y se abre la vista de crear actividad. Se llenan todos los campos y se sube al servidor un track de ruta vacío.	El sistema no debería registrar la nueva actividad y avisaría al usuario de que el fichero de track de ruta .gpx no contiene datos.
Caso de prueba 16	Crear actividad con un track de ruta inválido	Se accede al sistema como usuario registrado y se abre la vista de crear actividad. Se llenan todos los campos y se sube al servidor un fichero de ruta .gpx inválido, que no sigue su formato.	El sistema no debería registrar la nueva actividad y avisaría al usuario de que el fichero de track de ruta .gpx no se puede procesar.
Caso de prueba 17	Crear actividad subiendo un fichero de track de ruta con extensión (.exe)	Se accede al sistema como usuario registrado y se abre la vista de crear actividad. Se llenan todos los campos y se sube al servidor un fichero ejecutable .exe como track de la ruta.	El sistema no debería permitir subir como fichero de ruta un archivo ejecutable.

Caso de prueba 18	Crear una actividad sin guías	Se accede al sistema como usuario identificado y se accede al formulario de crear actividad. Se rellenan los campos obligatorios y se introduce un guía. Acto seguido se eliminan todos los guías y se trata de guardar la actividad nueva.	El sistema advertiría de que no se ha introducido ningún guía para la actividad y no la guardaría.
Caso de prueba 19	Crear una actividad de forma correcta	Se accede al sistema como usuario registrado y se abre la vista de crear actividad. Se rellenan todos los campos y se confirma la actividad nueva	El sistema debería registrar la nueva actividad y en la página principal debería figurar para la fecha de realización introducida.

4.6.1.8 CU-W-8 Editar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 20	Edición de actividad de en estado realizado.	Se accede al sistema como usuario registrado y se intenta editar una actividad realizada	El sistema no debería permitir acceder a la vista de edición de la actividad y avisando de que ya ha sido realizada.
Caso de prueba 21	Edición de actividad con campos vacíos	Se accede al sistema y se intenta editar una actividad dejando algún campo vacío.	El sistema no debería actualizar la actividad y avisaría con los campos que son inválidos.
Caso de prueba 22	Edición de los guías en una actividad en curso.	Se accede al sistema y se intenta editar una actividad en estado en curso. Se elimina un guía de la actividad y se guarda la edición	El sistema debería advertir al usuario que en una actividad en curso no se pueden eliminar guías ya existentes. No se habría guardado la edición de la actividad.
Caso de prueba 23	Edición de la fecha de realización a una fecha pasada	Se accede al sistema y se intenta editar una actividad en estado en curso. Se introduce una fecha de realización de la actividad menor a la fecha actual.	El sistema no debería permitir la edición de la fecha de realización a una fecha ya pasada.

4.6.1.9 CU-W-9 Cambiar datos personales

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 24	Edición incorrecta de contraseña	Se accede al sistema como usuario identificado y se accede al cambio de contraseña. Se introduce una nueva contraseña y se deja en blanco la repetición de la nueva contraseña. Se cierra sesión y se intenta entrar	El sistema no debería permitir finalizar la edición puesto que faltan datos por completar. Al intentar acceder al

		con la contraseña nueva que se introdujo en el primer campo.	sistema con la nueva contraseña el sistema debería notificar con credenciales inválidas al usuario y no permitir el acceso.
Caso de prueba 25	Edición correcta de contraseña	Se accede al sistema como usuario registrado y se edita la contraseña a '1234', '1234' en el campo de confirmación. Se cierra sesión y se vuelve a entrar con el login y la contraseña '1234'	El sistema debería entrar en sesión con la nueva contraseña.

4.6.1.10 CU-W-10 Monitorizar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 26	Monitorizar actividad sin excursionistas siendo monitorizados	Se accede a la vista de monitorización de una actividad en curso.	El sistema no debería mostrar en el mapa nada más que el track esperado para la ruta.
Caso de prueba 27	Mostrar trayectoria y posición y recargar con nuevos datos	Acceder a la monitorización de una actividad. Mostrar para dos excursionistas sus trayectorias y sus posiciones actuales. Esperar a que lleguen nuevos reportes para ambos excursionistas. Sin ocultar la trayectoria ni la posición, recargar las dos trayectorias de los excursionistas.	El sistema debería ocultar ambas polylíneas del mapa y la posición de ambos excursionistas y posteriormente recargar los datos de la tabla, actualizando los existentes, y automáticamente volver a mostrar ambas polylíneas pero con los datos actualizados.
Caso de prueba 28	Cambio de mapa con datos mostrándose	Mostrar mapa OSM. Mostrar la trayectoria de un excursionista y su posición actual. Cambiar el mapa a Raster.	La información del excursionista debería permanecer sobre el mapa en la misma posición y el mapa debería cambiar a Raster.
Caso de prueba 29	Monitorizar actividad con excursionistas sin reportes	Se accede a la vista de monitorización de una actividad en curso en la cual hay un excursionista, pero no ha enviado aún ningún reporte. Se intenta mostrar su trayectoria y posición actual.	El sistema no debería mostrar ninguna de los dos datos y no debería ni marchar el check en pantalla ni marcar como mostrada la trayectoria.

4.6.1.11 CU-W-11 Eliminar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 30	Prueba de imposibles: eliminar	Se accede a una actividad que esté en curso y creada por el propio usuario. Se intenta eliminar a través del acceso	El sistema debería mostrar inaccesible el

	actividad en curso	de los detalles	acceso a eliminar la actividad.
Caso de prueba 31	Acceso actividad eliminada a	Se accede a una actividad que esté pendiente de realización y creada por el propio usuario. Se elimina a través del acceso en los detalles de la actividad. Se vuelve a la lista de actividades y se busca la actividad que acaba de borrarse.	El sistema no debería listar la actividad borrada y por tanto sería inaccesible.

4.6.2 Subsistema aplicación móvil

4.6.2.1 CU-M-1 Registrarse

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 32	Registrarse sin conexión a internet en el dispositivo	Introducir credenciales nuevas y confirmar el registro	La aplicación debería avisar de que sin conexión a internet no se puede completar el registro.
Caso de prueba 33	Registro inválido por datos incompletos	Acceder al registro de la aplicación y llenar todos los campos menos el de correo electrónico.	La aplicación no debería permitir el registro del usuario avisando que faltan datos por llenar.
Caso de prueba 34	Registro válido	Acceder al registro de la aplicación y llenar todos los campos.	La aplicación deberá mandar la cuenta al servidor y vincularla con el dispositivo. Automáticamente deberá además ser redireccionado a la página principal

4.6.2.2 CU-M-2 Vincular cuenta

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 35	Vincular cuenta sin conexión a internet	Acceder a la aplicación sin conexión a internet, llenar credenciales de usuario e intentar vincular la cuenta con el dispositivo.	La aplicación deberá notificar al usuario que no hay conexión a internet y por tanto que no se puede realizar la vinculación.
Caso de prueba 36	Vincular cuenta inexistente en el sistema	Acceder a la aplicación e introducir unas credenciales no registradas en el sistema.	La aplicación deberá no vincular la cuenta de usuario.
Caso de prueba 37	Vincular cuenta de usuario correcta	Acceder a la aplicación e introducir de manera correcta las credenciales de una cuenta existente.	Vincular la cuenta de usuario con el dispositivo y mostrar la lista de actividades pendientes. Además, en sucesivas ejecuciones de la aplicación, ésta no deberá volver a

			preguntar por las credenciales de usuario.
Caso de prueba 38	Acceso con cuenta vinculada sin conexión a internet	Acceder a la aplicación e introducir de manera correcta las credenciales de una cuenta existente. Cerrar la aplicación y desconectar la conexión a internet. Acceder a la aplicación.	La aplicación debería iniciar en sesión al usuario advirtiéndole de que no tiene conexión y que los datos que se muestren pueden no ser los actualizados.

4.6.2.3 CU-M-3 Ver actividades

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 39	Ver actividades pendientes de realización	Acceder a la aplicación con una cuenta vinculada en el dispositivo.	La aplicación deberá mostrar en orden de fecha de realización las actividades disponibles en el sistema
Caso de prueba 40	Pérdida conexión internet y recarga de actividades	Acceder a la aplicación con una cuenta vinculada en el dispositivo. Desactivar la conexión a internet del mismo. Recargar las actividades.	La aplicación deberá avisar al usuario de que se ha perdido la conexión con internet y deberá almacenar de forma local las actividades descargadas hasta el momento. Mostrará las actividades de la última vez que se recargaron los datos y no permitirá recargarlos más hasta no tener conexión a internet de nuevo en el dispositivo.
Caso de prueba 41	Acceso a actividades sin conexión internet	Desactivar la conexión a internet del dispositivo y posteriormente acceder a la aplicación con una cuenta vinculada en el dispositivo.	La aplicación deberá avisar de que no hay conexión a internet y que por tanto se muestran las actividades descargadas en el último acceso con conexión.

4.6.2.4 CU-M-4 Comenzar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 42	Comenzar actividad sin conexión internet	Desactivar la conexión a internet del dispositivo y posteriormente comenzar una actividad para la que el usuario es guía. Recuperar la conexión y recargar la actividad	La aplicación deberá guardar en local el nuevo estado de la actividad y la contraseña introducida para iniciarla. Al recuperar la conexión y recargar los datos de la actividad, ésta debería estar en estado en curso.
Caso de prueba 43	Recuperar conexión	Desactivar la conexión a internet del dispositivo y	La aplicación deberá enviar al servidor el estado de la

	internet con actividad comenzada	posteriormente comenzar una actividad para la que el usuario es guía.	actividad y la contraseña almacenados en local y figurar en la lista de actividades como actividad en curso.
Caso de prueba 44	Contraseña de actividad inválida	Comenzar una actividad que esté pendiente de realización y dejar vacío el campo de contraseña.	La aplicación no deberá dar por comenzada la actividad y avisará al guía de la misma de que la contraseña introducida para la actividad es inválida.
Caso de prueba 45	Comenzar actividad con conexión a internet	Comenzar una actividad que esté pendiente de realización.	La aplicación deberá dar por comenzada la actividad y figurar accesible en la lista de actividades.

4.6.2.5 CU-M-5 Monitorizar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 46	Monitorizar actividad con conexión a internet	Acceder a la monitorización de una actividad.	La aplicación deberá mostrar los datos de todos los excursionistas siendo monitorizados. Además, deberá guardar de forma local los datos de posición de cada excursionista cada vez que éstos se recarguen.
Caso de prueba 47	Nuevo excursionista añadido a la actividad durante la monitorización.	Acceder a la monitorización de una actividad. Añadir un excursionista siendo monitorizado. Recargar los datos de la monitorización.	La aplicación deberá mostrar los datos de todas las personas que estaban siendo monitorizadas e incluir al nuevo excursionista.
Caso de prueba 48	Pérdida de conexión a internet monitorizando	Acceder a la monitorización de una actividad. Desactivar la conexión a internet del dispositivo. Recargar los	La aplicación deberá avisar al usuario de

	actividad	datos de la monitorización.	que la conexión a internet se ha perdido. Una vez recargada la actividad deberá recuperar y mostrar los datos almacenados de manera local.
Caso de prueba 49	Cierre de actividad durante la monitorización.	Acceder a la monitorización de una actividad. Cerrar la actividad pasándola a estado closed. Recargar los datos de la monitorización.	La aplicación deberá descargar el nuevo estado de la actividad del servidor y notificar al usuario de que ha sido cerrada.

4.6.2.6 CU-M-6 Unirse a actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 50	Unirse a actividad cerrada	Acceder a la aplicación como usuario identificado. Abrir el detalle de una actividad en curso. Cerrar la actividad pasándola a estado closed. Unirse a la actividad aportando la password de acceso.	La aplicación no deberá permitir unirse al usuario y mostrará al usuario una advertencia avisando de que la actividad no está en curso y por lo tanto no puede unirse a ella.
Caso de prueba 51	Unirse a actividad con contraseña inválida	Acceder a la aplicación como usuario identificado e introducir una contraseña errónea para una actividad	La aplicación no deberá permitir unirse al usuario y mostrará al usuario una advertencia avisando de que la actividad no está en curso y por lo tanto no puede unirse a ella.
Caso de prueba 52	Pérdida de cobertura de datos durante una actividad estando monitorizado	Acceder a la aplicación como usuario identificado. Unirse a una actividad aportando para ello la contraseña de la misma. Desconectar la conexión a internet del dispositivo. Esperar a la siguiente actualización de datos de posición.	La aplicación deberá avisar al usuario de la pérdida de conexión a internet. La aplicación deberá obtener la posición GPS del dispositivo y almacenarla de manera local.

Caso de prueba 53	Reconexión a internet con reportes de posición almacenados de manera local estando monitorizado en una actividad.	Acceder a la aplicación como usuario identificado. Unirse a una actividad aportando para ello la contraseña de la misma. Desconectar la conexión a internet del dispositivo. Esperar a la siguiente actualización de datos de posición. Reconectar el dispositivo a internet.	La aplicación deberá avisar al usuario de la reconexión a internet. Enviará al servidor y eliminará posteriormente todos los datos de posición almacenados de manera local. Obtendrá la posición GPS del dispositivo y la enviará al servidor.
Caso de prueba 54	Envío de reporte de situación con conexión a internet	Acceder a la aplicación como usuario identificado. Unirse a una actividad aportando para ello la contraseña de la misma. Esperar a la siguiente actualización de datos de posición.	La aplicación obtendrá la posición GPS del dispositivo y la enviará al servidor.
Caso de prueba 55	Cierre de actividad estando monitorizado	Acceder a la aplicación como usuario identificado. Unirse a una actividad aportando para ello la contraseña de la misma. Cerrar la actividad. Esperar a la siguiente actualización de datos de posición.	La aplicación deberá descargar el nuevo estado de la actividad y descartar, en caso de haberlos, reportes almacenados de manera local. Notificará mediante una advertencia en la pantalla al usuario de que la actividad ha sido cerrada y la monitorización ha concluido.
Caso de prueba 56	Cierre de actividad estando monitorizado sin tener conexión	Acceder a la aplicación como usuario identificado. Unirse a una actividad aportando para ello la contraseña de la misma. Desactivar la conexión a internet del dispositivo. Cerrar la actividad. Esperar a la siguiente actualización de datos de posición. Recuperar la conexión a internet.	El sistema debería desechar los reportes de posición enviados tras el cierre de la actividad y debería notificar al usuario de que la actividad ha sido concluida.

4.6.2.7 CU-M-7 Finalizar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 57	Finalizar actividad sin cobertura de datos	Acceder a la aplicación como guía de una actividad en curso. Desactivar la conexión a internet del dispositivo. Finalizar la actividad.	La aplicación deberá guardar de manera local el estado nuevo de la actividad y notificar al usuario de que se ha perdido la conexión a

			internet y por tanto no ha podido ser enviado al servidor.
Caso de prueba 58	Finalizar actividad con cobertura de datos	Acceder a la aplicación como guía de una actividad en curso. Finalizar la actividad.	La aplicación deberá enviar al servidor el nuevo estado de la actividad. Notificará al usuario de que la actividad ha sido finalizada.

4.6.2.8 CU-M-8 Cambiar password

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 59	Cambio de contraseña sin conexión a internet	Acceder a la aplicación a la vista de datos de cuenta. Desactivar la conexión a internet del dispositivo. Cambiar la contraseña	La aplicación deberá notificar tras el tiempo establecido de timeout que el cambio de contraseña no se ha podido llevar a cabo.

4.6.2.9 CU-M-9 Abandonar actividad

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 60	Abandono de actividad con reportes acumulados de manera local y sin conexión a internet	Unirse a una actividad sin conexión a internet. Esperar a que se acumulen tres reportes en local. Abandonar la actividad. Reactivar la conexión a internet del dispositivo	El sistema debería realizar las acciones que se llevaron a cabo sin conexión en el mismo orden. Deberían recogerse y registrarse todos los reportes enviados por el usuario y posteriormente dejar de ser monitorizado al abandonar la actividad

4.6.2.10 CU-M-10 Cerrar sesión

Identificador	Descripción	Procedimiento	Salida esperada
Caso de prueba 61	Cierre de sesión con vinculación de cuenta guardada	Iniciar sesión y recordar los datos. Cerrar sesión y reiniciar la app.	No debería iniciar sesión al usuario y por tanto se quedaría en la pantalla de login.

Capítulo 5. Diseño del Sistema

5.1 Especificación y justificación de la plataforma tecnológica utilizada

La arquitectura que se quiere lograr para el desarrollo de la aplicación es una arquitectura basada en microservicios. En dicha arquitectura, cada elemento de la aplicación, servidor y clientes, tendrían responsabilidades únicas y serían totalmente independientes.

Para lograrlo se ha hecho un estudio de posibles tecnologías, teniendo en cuenta diferentes opciones, y seleccionando aquellas que permitiesen una mejor adaptación al modelo de aplicación que se quería conseguir.

A continuación, se explicarán cada una de las tecnologías seleccionadas para implementar la aplicación.

5.1.1 Spring Framework

Se trata de un framework para el desarrollo de aplicaciones, que es de código abierto y diseñado para la plataforma Java.



Figura 78 Logo Spring Framework

Una de las principales características de Spring es la inyección de dependencias (IoC), la cual consiste en la gestión de objetos java utilizando la reflexión. Es el propio contenedor el encargado de gestionar todos estos objetos, desde la instancia hasta la configuración de los mismos.

Con Spring se pueden construir de manera sencilla aplicaciones para ofrecer servicios web RESTful y de esta forma conseguir una arquitectura basada en microservicios.

Por todas estas características, entre otras, se ha seleccionado Spring Framework como tecnología con la que construir el servidor de la aplicación, el cual será el encargado de gestionar todos los datos aportados y consumidos por los diferentes clientes.

5.1.2 Sistema Gestor de Base de Datos utilizado

Gracias a la versatilidad de configuración que aporta Spring Framework, dependiendo del tipo de despliegue que se realice con la aplicación, sin alterar el código del servidor se puede configurar el despliegue para hacerlo en MySQL o PostgreSQL.

5.1.2.1 MySQL

Se trata de un sistema de gestión de bases de datos relacional que cuenta con multihilo desarrollado por Oracle como software libre. Algunas de sus principales características son:

- Cuenta con un amplio subconjunto de lenguaje SQL.
- Está disponible en una gran cantidad de plataformas y sistemas diferentes.
- Cuenta con transacciones y claves foráneas

5.1.2.2 PostgreSQL

Se trata de un sistema de gestión de base de datos relacional y orientada a objetos. Es software libre y su desarrollo está dirigido por toda una comunidad que trabaja de forma desinteresada para mejorarlo. Entre otras, sus principales características son:

- Alta concurrencia
- Arrays
- Texto largo ilimitado

5.1.3 Apache Tomcat

Contenedor de servlets que implementa especificaciones de JSP y es utilizado para desplegar el servidor.

5.1.4 Android

La utilización de Android como tecnología para el desarrollo del cliente móvil no fue la primera opción contemplada. A continuación, se expone la primera opción, la utilización de un framework multiplataforma, por qué fue descartada y las ventajas de Android.

5.1.4.1 Utilización de framework multiplataforma

La primera intención para el desarrollo del cliente móvil era hacerlo sobre un framework multiplataforma para no restringir el uso de la aplicación a determinado tipo de dispositivos móviles. El framework que fue elegido para su utilización fue Ionic.



Figura 79 Logo Framework Ionic

Esta idea tuvo que desecharse debido a las tareas que se querían mantener en segundo plano. A través de la API de Android (veremos sus ventajas más adelante) se podía conseguir la automatización de tareas en segundo plano, de manera estable y garantizada en periodos de tiempo, incluso cuando el dispositivo no tenía la aplicación dispuesta en primer plano. Esto no pasaba con Ionic, además de que requería un periodo de formación adicional.

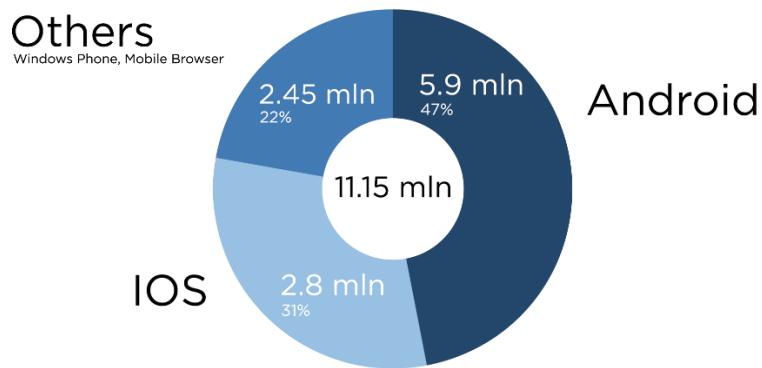
La ejecución de dichas tareas en segundo plano es uno de los pilares fundamentales de la aplicación puesto que es a través de la cual se recogen los datos de monitorización durante las actividades que gestionará la aplicación.

Por ello, se decidió descartar esta alternativa y utilizar Android, el cual presentaba otro tipo de ventajas salvando la de ejecución en dispositivos multiplataforma.

5.1.4.2 Ventajas de Android

Se ha elegido Android como la tecnología a emplear para el desarrollo del cliente móvil de la aplicación. Han sido varios los motivos por los cuales se ha llegado a esta decisión:

- API Android. La API de Android está perfectamente documentada y actualizada. A través de ella se puede acceder a cualquier servicio de manera rápida, fácil, sencilla y optimizada que ofrezca el dispositivo que instale la aplicación. Tiene una comunidad de desarrolladores detrás significativa lo que conlleva que sea sencillo encontrar solución a diferentes problemas que puedan surgir durante el desarrollo o comprensión de la misma.
- Ejecución de tareas en background. Como se explicó anteriormente, la facilidad de llevar a cabo tareas de manera eficaz y estable con Android fue uno de los motivos más importantes para la selección como tecnología a utilizar.
- Librerías de terceros. Son muchas las desarrolladas para implantarse directamente sobre aplicaciones desarrolladas en Android.
- Lenguaje de programación. Java es uno de los lenguajes disponibles y el principal para el desarrollo de aplicaciones Android. Esto es una ventaja sobre cualquier otro framework que utilice otro lenguaje.
- Utilización de Android como sistema operativo frente a otros del mercado. Actualmente Android sigue siendo líder en sistema operativo en smartphones del mercado.



Tal y como se observa en esta gráfica de 2017, Android cuenta con casi la mitad del mercado, frente a sus principales competidores como por ejemplo, iOS.

5.1.5 Javascript

Se trata de un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico

Es el lenguaje de programación utilizado para el desarrollo del cliente web.

5.1.6 Angular JS

Se trata de un framework de Javascript de código abierto y mantenido por Google que es utilizado para la construcción de aplicaciones web de una sola página, esto es, en ningún momento el navegador refresca o carga otra página, sino que es el framework el que realiza este trabajo.



Figura 80 Logo Angular JS

Está orientado a servir mejor el contenido dinámico y a enlazar de manera bidireccional los datos de la vista con el modelo.

5.2 Arquitectura del sistema

5.2.1 Patrones de diseño utilizados

En este apartado de la arquitectura del sistema se introducirán aquellos patrones de diseño más significantes que se han utilizado para el desarrollo tanto de la aplicación móvil, aplicación web como para el servidor y que han servido para solucionar de una manera eficiente determinados problemas o situaciones a la hora de desarrollar cierta funcionalidad de la aplicación.

5.2.1.1 N-Capas

El patrón N-Capas se ha implementado en el lado servidor. Las principales características de una arquitectura basada en capas son las siguientes:

- Cada capa conoce y realiza sus responsabilidades, diferentes de las del resto de capas.
- Cada capa solo se comunica con la inmediatamente sucesora a ella
- Cada capa solo conoce de su sucesora aquellos métodos que va a necesitar

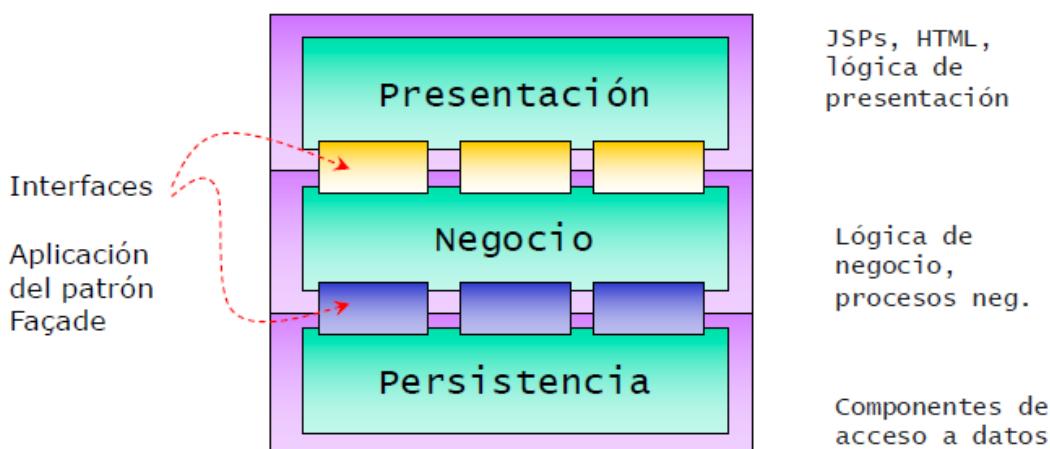


Figura 81 Esquema modelo de Brown N-Capas

Las diferentes capas son:

- Presentación: contiene la lógica de la vista, la que proyecta los datos ya preparados para ello. No tiene lógica de datos, la información ya viene procesada de negocio.
- Negocio: servicios de negocio de la aplicación. La presentación consume los datos de esta capa a través de las fachadas pasando los datos (o no) que son recogidos en la vista y aportados por el usuario. Estos datos son procesados y devueltos para ser presentados en la capa de presentación.
- Persistencia: la persistencia de los datos que consume y envía la capa de negocio. Si se cambiase el SGDB del sistema, la única capa que se debería tocar sería la de Persistencia, quedando Negocio y Presentación sin alteraciones. Es la que se

encarga de persistir y recuperar los datos para ser enviados a negocio cuando sean solicitados.

Gracias a las fachadas, la capa cliente, entendiéndose por esto la que va a llamar a su sucesora, solo conoce de ésta lo imprescindible.

El modelo N-Capas nos deja como beneficios un mayor aislamiento del código, esto es, para realizar un cambio en negocio no hará falta modificar la presentación o la persistencia mientras que sus fachadas no se vean alteradas.

Por otro lado, una aplicación implementada bajo el patrón N-Capas da por resultado un código mucho más fácil de mantener (por lo citado anteriormente), robusto y eficiente.

5.2.1.2 *Modelo vista controlador (MVC)*

Se trata de un patrón de arquitectura que separa los datos de la lógica de negocio, de la presentación y del encargado de gestionar los eventos y comunicaciones.

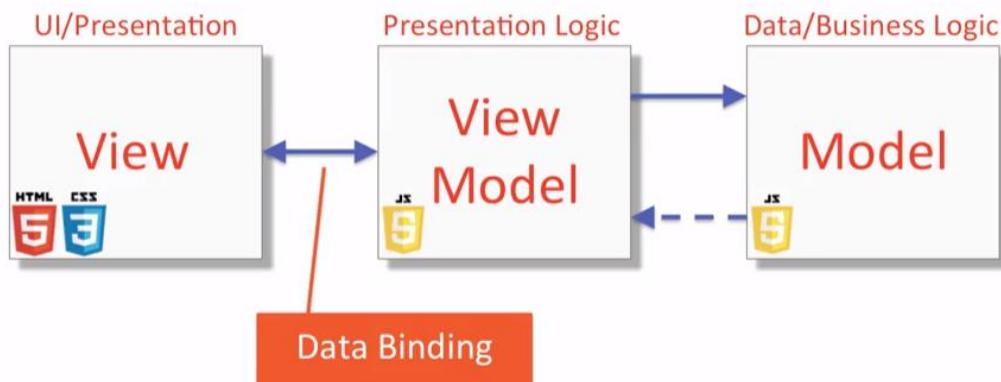
Para conseguir este objetivo se plantea la separación en tres componentes totalmente diferenciados:

- Modelo: la información de la aplicación, los datos. Gestiona todos los accesos a esa información.
- Vista: presenta la información del modelo en el formato requerido.
- Controlador: responde a eventos e invoca las peticiones al modelo. Se puede decir que interactúa como intermediario entre la vista y el controlador.

5.2.1.3 *Model View ViewModel*

Se puede entender como una derivación o especificación del MVC. Es el utilizado en la aplicación web a través de AngularJS.

Model-View-ViewModel



El Model View ViewModel cuenta con tres elementos: view, model, y viewModel. Cada uno de ellos tiene sus responsabilidades separadas del resto.

- View: se puede entender como todo el código HTML y CSS de la web, es decir, la presentación. Consumo los datos ya procesados, nunca los manipula, y los muestra al usuario.
- Model: solo los datos de la aplicación, los almacena. Además, también contiene toda la lógica. Esto es, por ejemplo, desde el modelo se llama a un servicio HTTP para obtener unos datos, se formatean de manera adecuada y son enviados a la vista para ser representados al usuario.
- ViewModel: representa el estado de la vista. Se puede entender como la lógica de la presentación, es decir, el código JavaScript escrito en presentación. Contiene aquellos datos seleccionados por el modelo para ser mostrados en la vista y responde a eventos sobre ellos.

Todo este esquema no tiene sentido sin el concepto de “Declarative Binder”: se encarga de comunicar la View con ViewModel. Que sea declarative indica que esto es conseguido por el propio framework (Angular JS) y que por tanto esos bindings bidireccionales explicados anteriormente no son responsabilidad del programador, sino que se proporcionan automáticamente.

Esto garantiza en el subsistema de la aplicación web que el modelo interactúe a través de eventos y llamadas con la vista gracias al ViewModel y, por ende, al data binding.

5.2.1.4 *Modelo Vista Presentador (MVP)*

Se considera una derivación del patrón MVC, utilizado mayoritariamente para la construcción de interfaces de usuario.

Uno de sus principales objetivos es la separación de la lógica de negocio de la lógica de presentación. Está compuesto por modelo, vista y presentador:

- Modelo: define la interfaz de los datos
- Presentador: actúa sobre el modelo y la vista. Recupera los datos del modelo y los formatea mediante la lógica de negocio para ser mostrados al usuario en la vista. Actúa como mediador entre el modelo y la vista, de esta forma se consigue que no se conozcan entre ellos.
- Vista: interfaz para mostrar los datos al usuario.

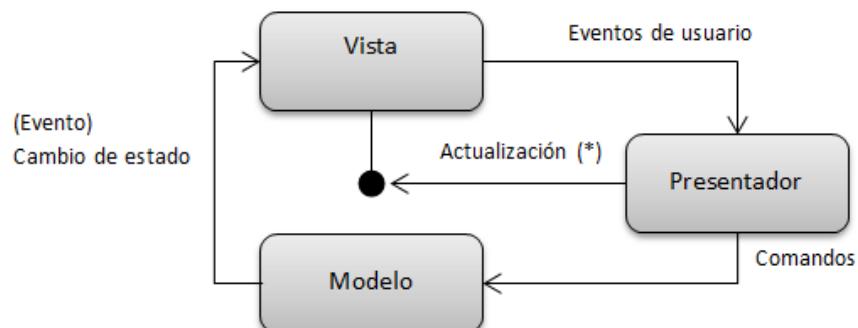


Figura 82 Estructura patrón Modelo Vista Presentador (MVP)

Una de las aplicaciones más utilizadas del patrón MVP es en aplicaciones Android. Mediante la aplicación de este patrón se consigue un código mucho más limpio, robusto y escalable.

A continuación, se ilustra la aplicación de este patrón con el diagrama de clases de la activity de configuración de la aplicación Android.

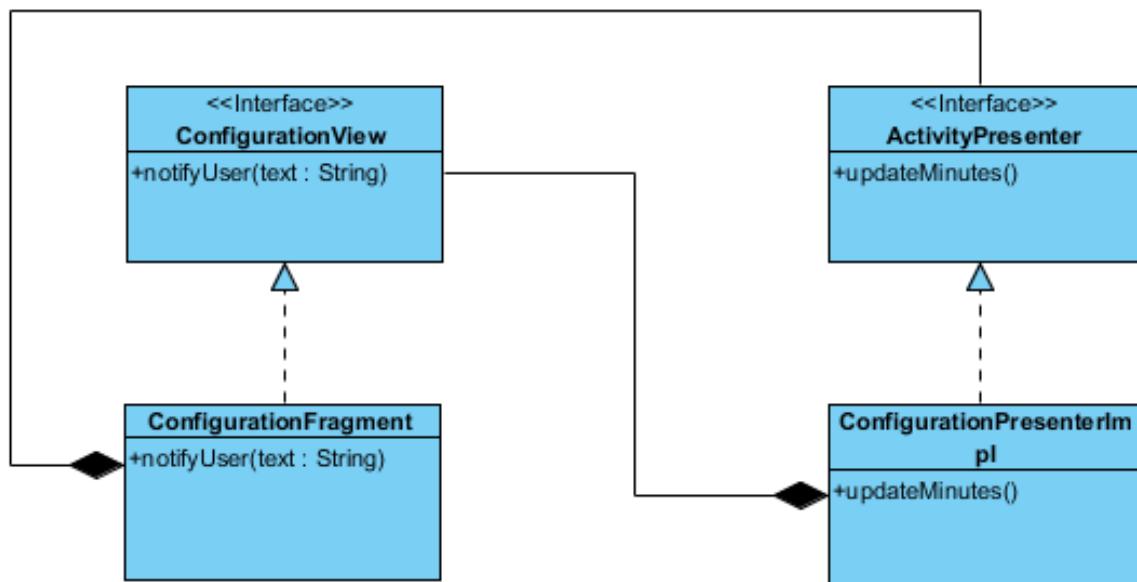


Figura 83 Diagrama de clases Activity Configuración – aplicación Android

Como se puede observar en el diagrama de clases anterior, el fragment `ConfigurationFragment` implementa la interface `ConfigurationView`, que es la única que conoce el presentador. De esta manera, éste, solo puede interactuar con la vista a través del método `notifyUser` sin conocer nada más a cerca de esta clase.

5.2.1.5 *Strategy*

Se trata de un patrón de comportamiento que establece qué hacer para resolver una tarea en concreto. En otras palabras, lo aplicamos cuando para hacer la misma tarea tenemos diferentes versiones, formas o algoritmos diferentes.

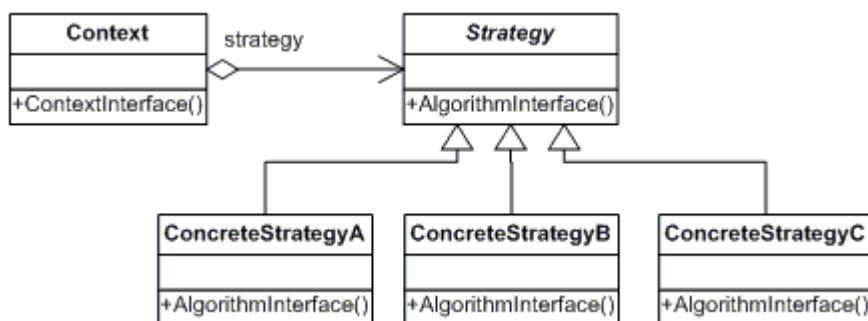


Figura 84 Estructura patrón strategy

El patrón strategy ha sido utilizado en más de un punto de la aplicación para resolver diferentes problemas, pero se puede destacar su uso en la aplicación móvil. Es utilizado con las diferentes formas que se tiene para representar el mapa.

Para mostrar el mapa se puede utilizar cualquier proveedor. Esto se ha conseguido a través del patrón strategy, de tal manera que cada estrategia concreta es un tipo de mapa.

A continuación, junto con la aplicación del patrón Singleton, se muestra un diagrama de clases ilustrando este comportamiento con los tipos de mapa de la aplicación.

5.2.1.6 *Singleton*

El patrón de diseño Singleton tiene por objetivo garantizar que un objeto es instanciado nunca o una sola vez como máximo en todo el ciclo de vida de la aplicación.

Si nos interesa guardar en memoria algún dato sin necesidad de persistirlo, gracias al patrón singleton garantizamos esta funcionalidad.

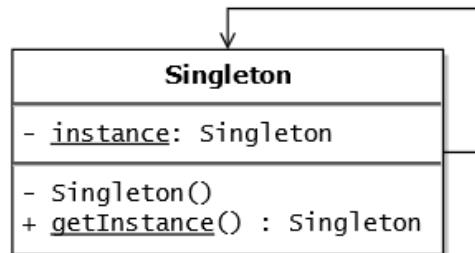


Figura 85 Estructura patrón Singleton

De la misma forma que con el patrón Strategy, Singleton también ha sido usado en diferentes puntos de la aplicación. Uno reseñable relacionado con el Strategy es la clase MapTypesManager, encargada de gestionar las instancias de los diferentes tipos de mapas y devolverlas cuando son necesarias.

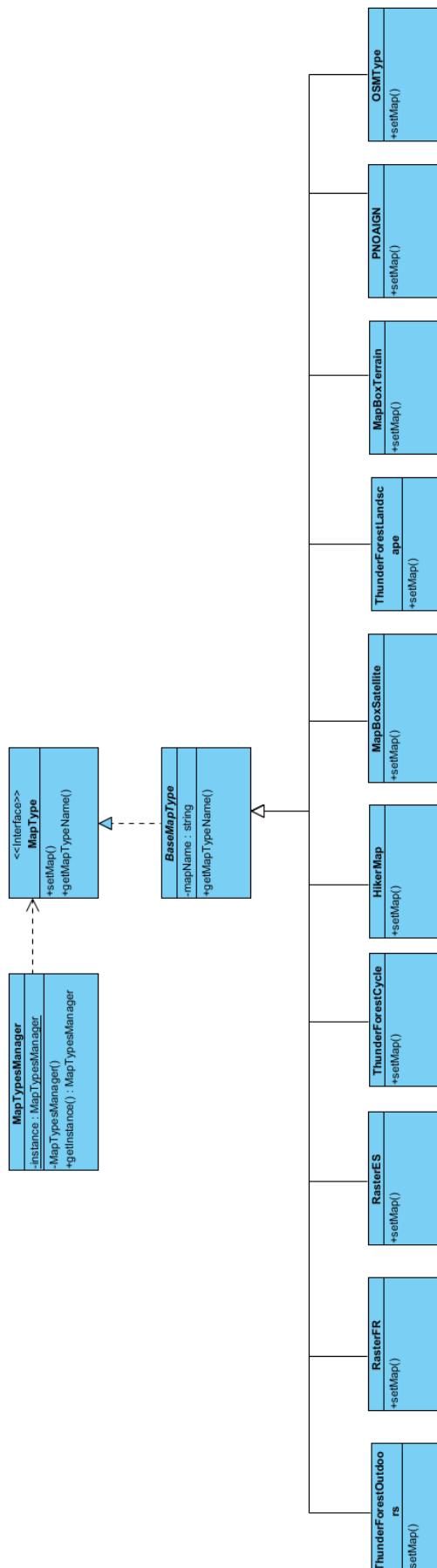


Figura 86 Diagrama de clases gestión de tipos de mapas – Aplicación Android

5.2.1.7 State

El patrón de diseño state permite a un objeto cambiar su comportamiento cuando cambia el estado interno de dicho objeto. Se utiliza cuando el comportamiento de un objeto depende de su estado.

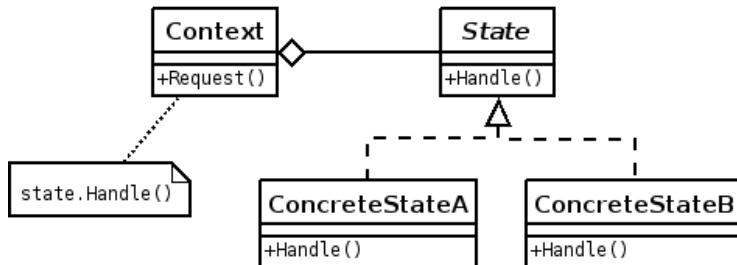


Figura 87 Estructura patrón state

Tal y como se puede apreciar en la estructura de clases del patrón state, tiene una implementación muy similar al strategy. Se pueden definir como que el patrón strategy define como hacer algo y el patrón state define que hacer.

El patrón state ha sido implementado en la aplicación Android para gestionar la forma de actuar en caso de estar en estado desconectado o conectado a la hora de gestionar un reporte de posición. La acción es la misma, gestionar el reporte de posición, pero se debe definir qué hacer en función de si hay o no conexión.

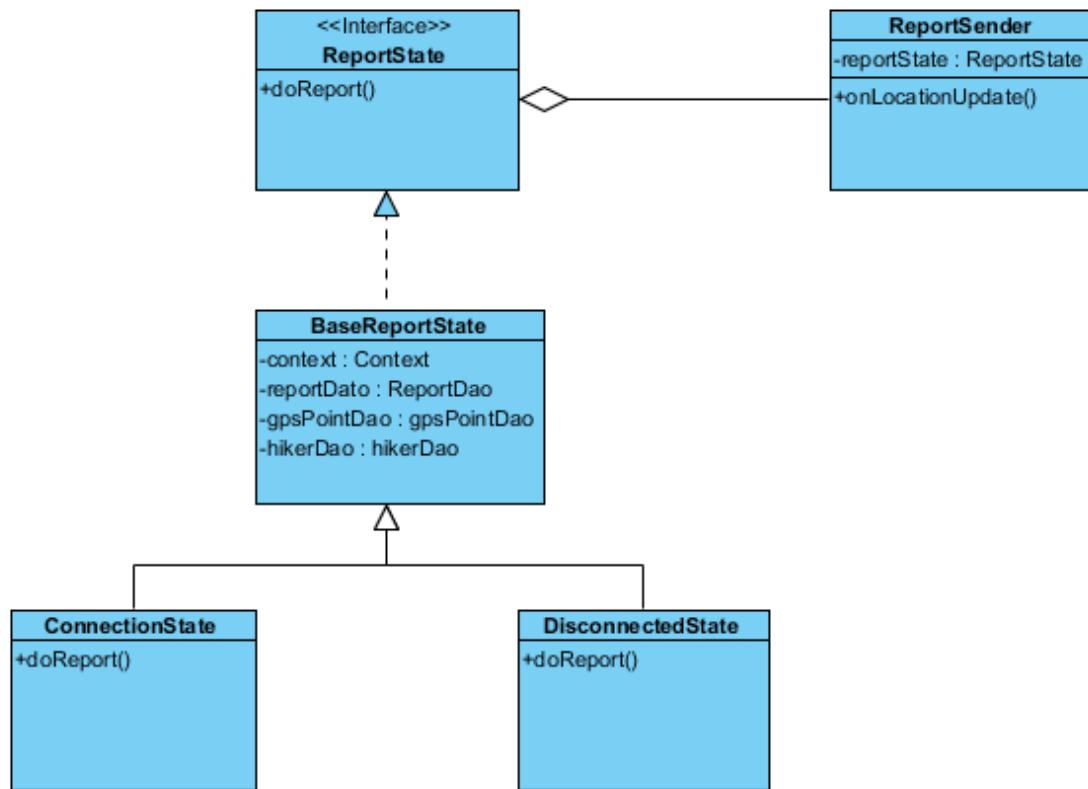


Figura 88 diagrama de clases envío de reportes – Aplicación Android

5.2.2 Diagrama de despliegue

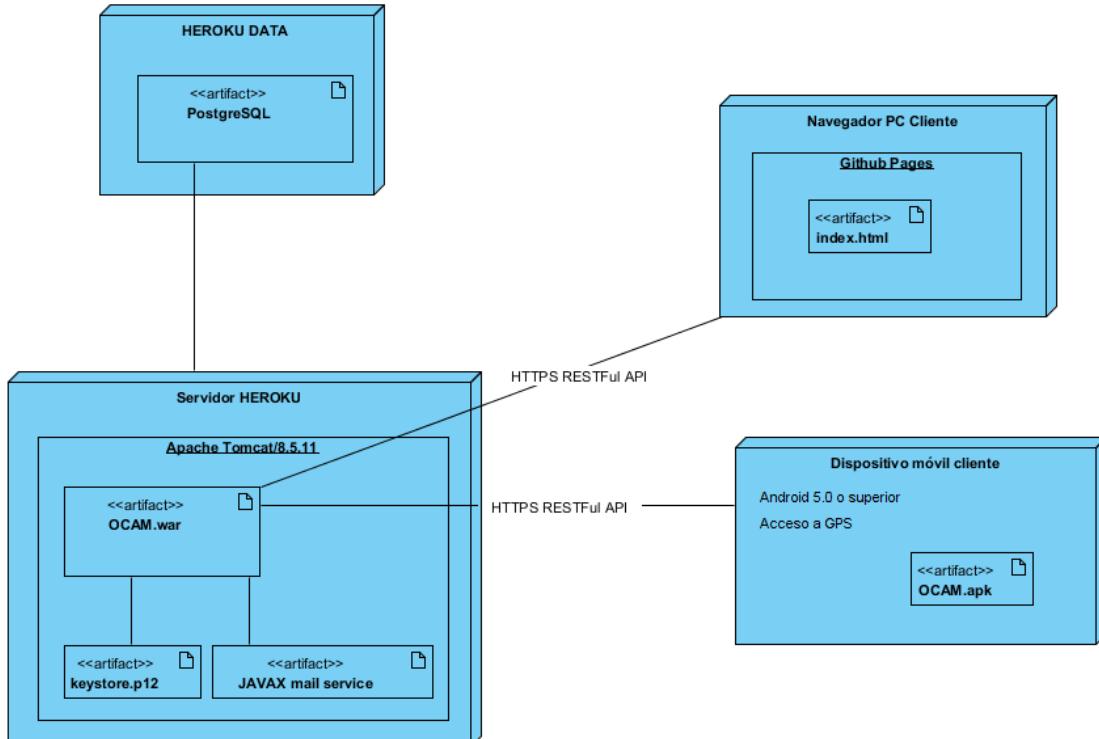


Figura 89 Diagrama de despliegue

- PostgreSQL: Base de datos PostgreSQL utilizada como sistema gestor de base de datos de la aplicación para la persistencia de la información.
- OCAM.war: empaquetación del servidor. Ofrece una API RESTFul para que los clientes puedan consumir los datos y enviarlos al propio servidor.
- Keystore.p12: certificado HTTPS.
- JAVA mail service: servicio de mail de Java para enviar emails.
- Index.html: página web del cliente aplicación web.
- OCAM.apk: aplicación móvil nativa Android cliente.

5.3 Autenticación de usuarios

5.3.1 JSON Web Tokens

Para la autenticación de los clientes contra el servidor, tanto en la aplicación web, como en la aplicación Android, en OCAM se utilizan JSON Web Tokens.

Se trata de una autenticación *stateless*. El funcionamiento básico es, a través de una API RESTful sin estado, un cliente, aportando unas credenciales de acceso (en el caso de OCAM), el servidor responderá a la petición con una cadena de texto encriptada, la cual contendrá información del usuario y una fecha de caducidad (el token). En sucesivas peticiones, el cliente, se identificará mandando en la cabecera de las mismas el token que el servidor le ha aportado. Cuando el token caduque, será el servidor el que lo renovará ampliando su fecha de caducidad.

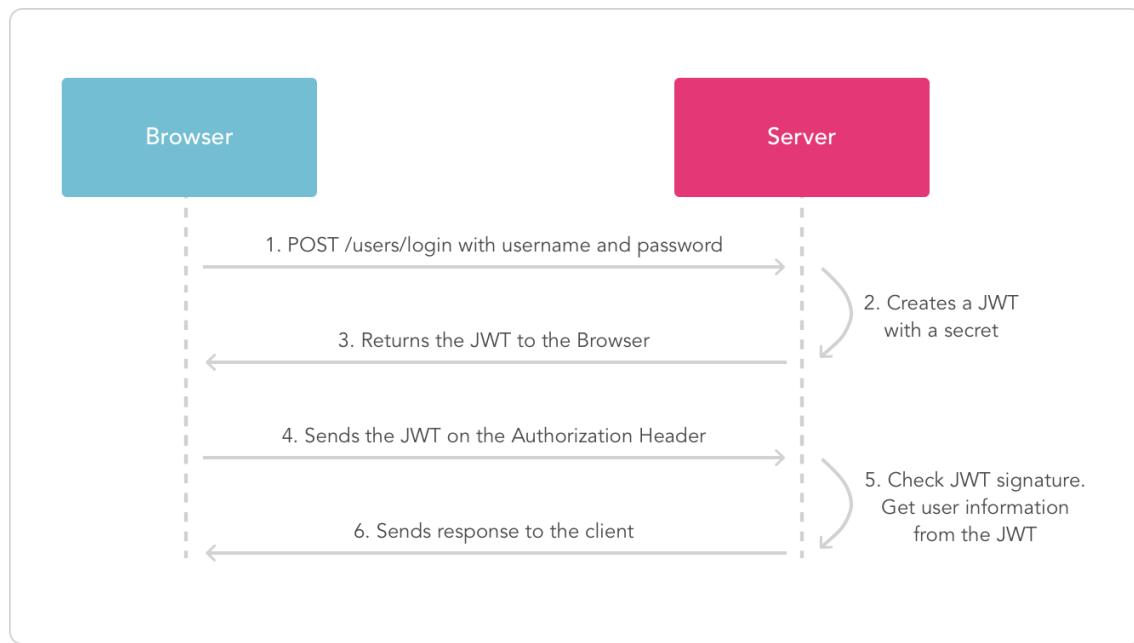


Figura 90 Esquema de funcionamiento de Autenticación por token

Un token es una cadena de caracteres, compuesta a su vez por tres subcadenas que están divididas por un punto.

- Header: la primera parte del token que contiene la información del tipo de token, y el algoritmo de encriptación del mismo. En el caso de la aplicación el tipo de token es JWT y se usa el algoritmo de encriptación HS256.
- Payload: el contenido del token. Dependiendo de las necesidades de la aplicación contendrán unos parámetros u otros. En OCAM, el payload del token contiene la fecha en la que caduca y los datos del usuario (login y correo electrónico) dueño de dicho token.
- Signature (firma): cadena encriptada en Base64 formada por la concatenación del Header y el Payload del propio token.

Un ejemplo de token podría ser el siguiente:

Header: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
 Payload: eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjItOGZhYi1jZWYzOTA0NjYwYmQifQ.
 Signature:-xN_h82PHVTCMA9vdoHrcZxH-x5mb1ly1537t3rGzcM

El beneficio de uso que aporta la autenticación por medio de JSON Web Tokens es que, entre otras cosas, al estar los tokens almacenados en el cliente, la aplicación se vuelve totalmente escalable.

Teniendo un servidor podemos conectar tantos clientes como queramos a través de la API RESTful sin que el servidor sepa en ningún momento que tipo de aplicación implementan dichos clientes.

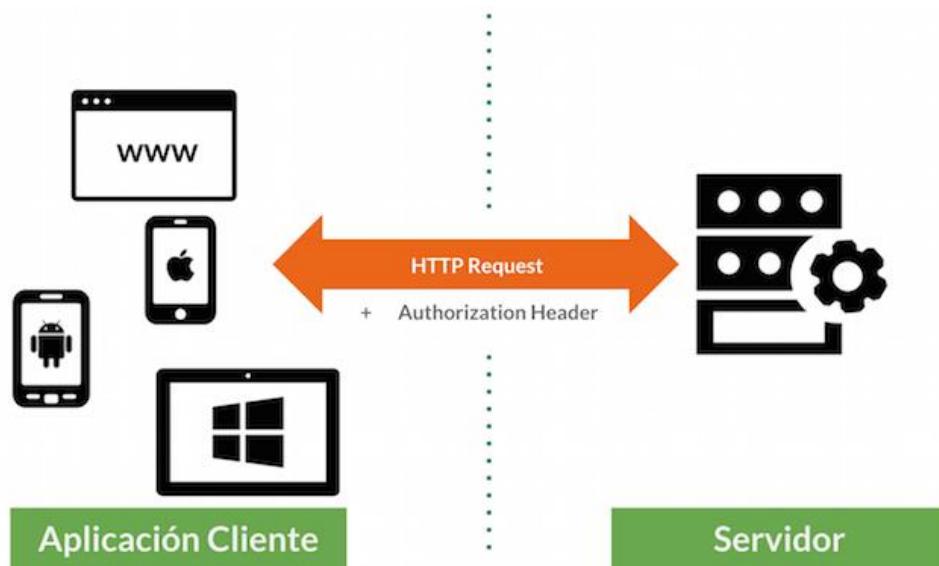


Figura 91 Esquema de funcionamiento de JWT con varios clientes

Otro aspecto importante a resaltar es la seguridad que aporta este tipo de autenticación. Al no haber que mantener sesiones con el servidor, éstas, no pueden ser manipuladas de ninguna manera.

5.3.1.1 Comunicaciones clientes-servidor

Para la realización de las comunicaciones entre los clientes y el servidor se utiliza el protocolo HTTPS mediante un certificado en el servidor. Esto garantiza que las comunicaciones son seguras.

La única comunicación que mueve datos sensibles es en la que el cliente solicita el token al servidor por primera vez. En esa petición HTTP viajan en la cabecera de la misma las credenciales del usuario. Para incrementar la seguridad, dichas cabeceras viajan encriptadas en Base64 además de, como se ha explicado ya, realizarse a través de HTTPS.

5.3.2 Mantenimiento de la sesión

Para mantener la sesión en la aplicación web se utiliza la librería de Angular LocalStorage. Básicamente se trata de una implementación de WebStorage en AngularJS y compatible con cualquier aplicación web desarrollada utilizando este framework.

Las principales diferencias con Angular Session-Storage es que los datos que almacena esta otra opción son temporales.

Una de las principales diferencias con otros métodos de almacenamiento de información en el navegador es la seguridad. En el caso de Angular Local-Storage, es posible acceder a la información que almacena. En el caso de OCAM esto no es un problema.

La información que se guarda de un usuario es el nombre de usuario, el email, el token y el token de refresco (utilizado este segundo para aumentar el tiempo de vida del primero).

En ningún momento, aunque se acceda al token se podría sacar información sensible, puesto que éste no almacena más que email, nombre de usuario y fecha de caducidad, además de estar encriptado.

En el caso de la aplicación móvil, para mantener las cuentas vinculadas sin necesidad de tener conexión a internet se utilizan las *SharedPreferences* privadas de la aplicación. En ellas se guarda la misma información que en la aplicación web, es decir, nombre de usuario, correo electrónico y token normal y de refresco.

A continuación, en los diagramas de interacción de casos de uso se expone el diagrama de login en la aplicación, el cual ilustra todo el proceso de llamadas hasta obtener un token.

5.4 Diagramas de interacción de casos de uso

En este apartado del diseño se expondrán los diagramas de interacción de los casos de uso que se han identificado como más relevantes en cada uno de los subsistemas de la aplicación.

5.4.1 Identificación en el sistema

Tal y como se explicó en el apartado anterior, la autenticación en el sistema se realiza por medio de tokens. En el siguiente diagrama de interacción se ilustra como un usuario desde que hace la petición, se le genera un token y luego se le valida en sucesivas llamadas.

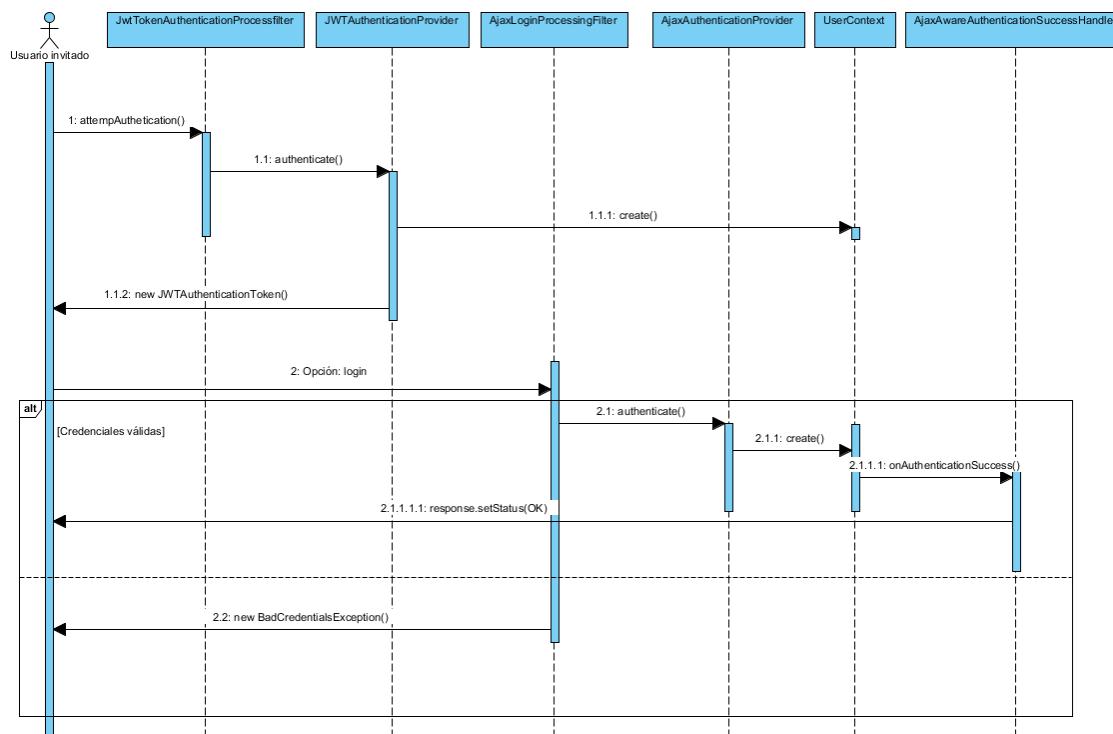


Figura 92 Diagrama de interacción – Autenticación servidor

5.4.2 Ver actividades

Al entrar un usuario, ya sea bajo el rol de usuario identificado o usuario invitado en la página web, puede acceder a la lista de actividades.

La aplicación web se descarga las actividades del servidor y se las dispone al usuario de manera que en cualquier momento puede recargarlas, filtrarlas u ordenarlas. Además, también se pueden filtrar según actividades realizadas o pendientes de realización.

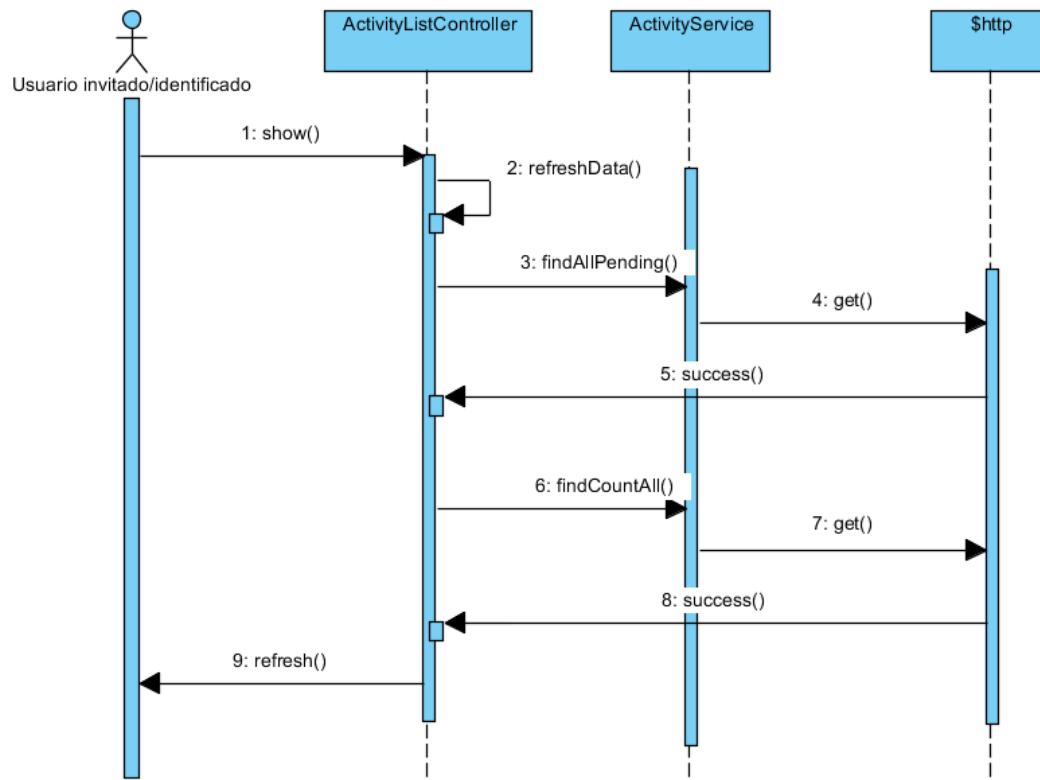


Figura 93 Diagrama de interacción - Ver actividades

5.4.3 Iniciar actividad

Desde el fragment de una actividad en la aplicación móvil, entre otras acciones, el usuario que es guía de la actividad puede darla por comenzada.

Al dar por comenzada la actividad la aplicación actualiza su estado en la memoria del dispositivo, hace una petición POST al servidor para actualizarlo y al recibir la respuesta del mismo

Por último, se inicia el fragment con la monitorización de la actividad y se comienza el Broadcast que recogerá de manera periódica la posición del dispositivo.

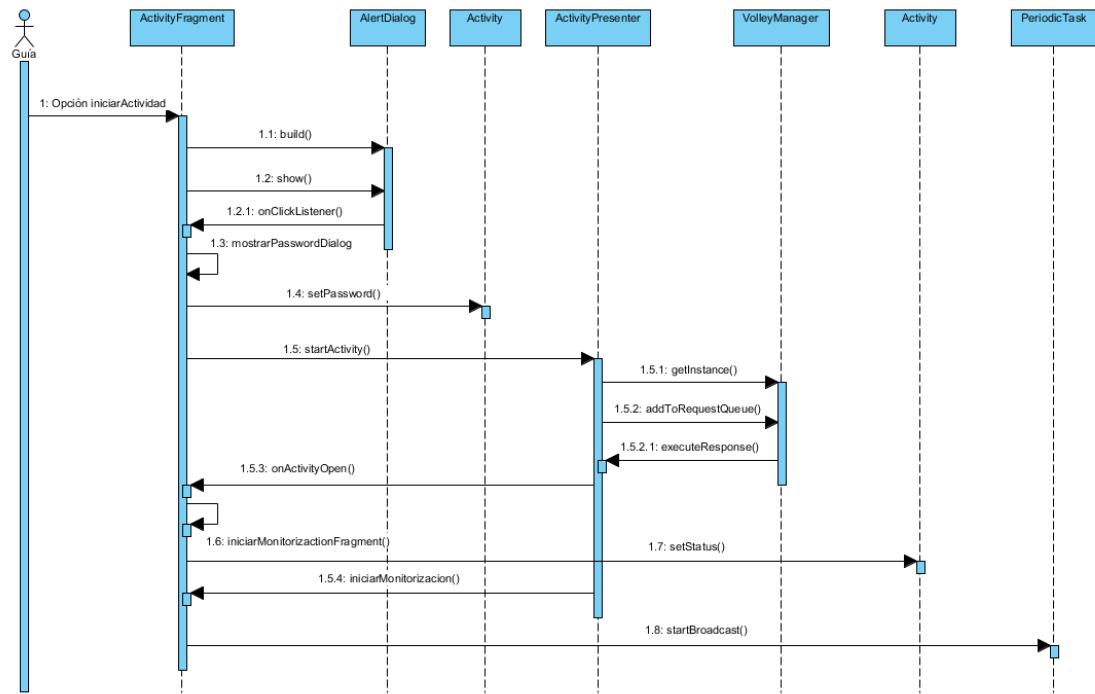


Figura 94 Diagrama de interacción – Iniciar actividad

5.4.4 Abandonar actividad

Desde el fragment de una actividad en la aplicación móvil, entre otras acciones, un usuario que está unido en una actividad y que además ésta está en curso, puede abandonarla.

Al abandonar la actividad se detiene la monitorización para el dispositivo y se manda una petición POST al servidor actualizando la actividad.

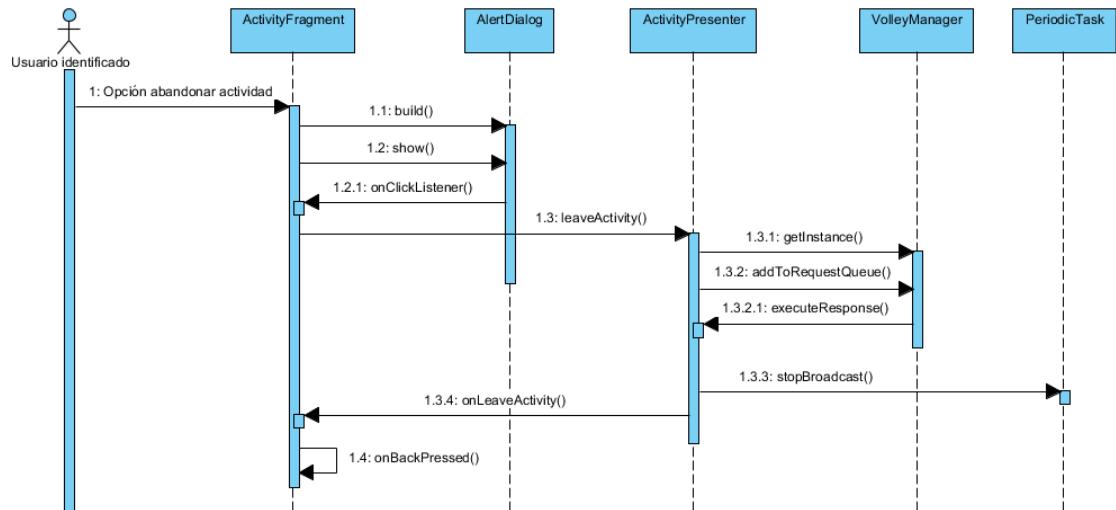


Figura 95 Diagrama de interacción – Abandonar actividad

5.4.5 Unirse a actividad

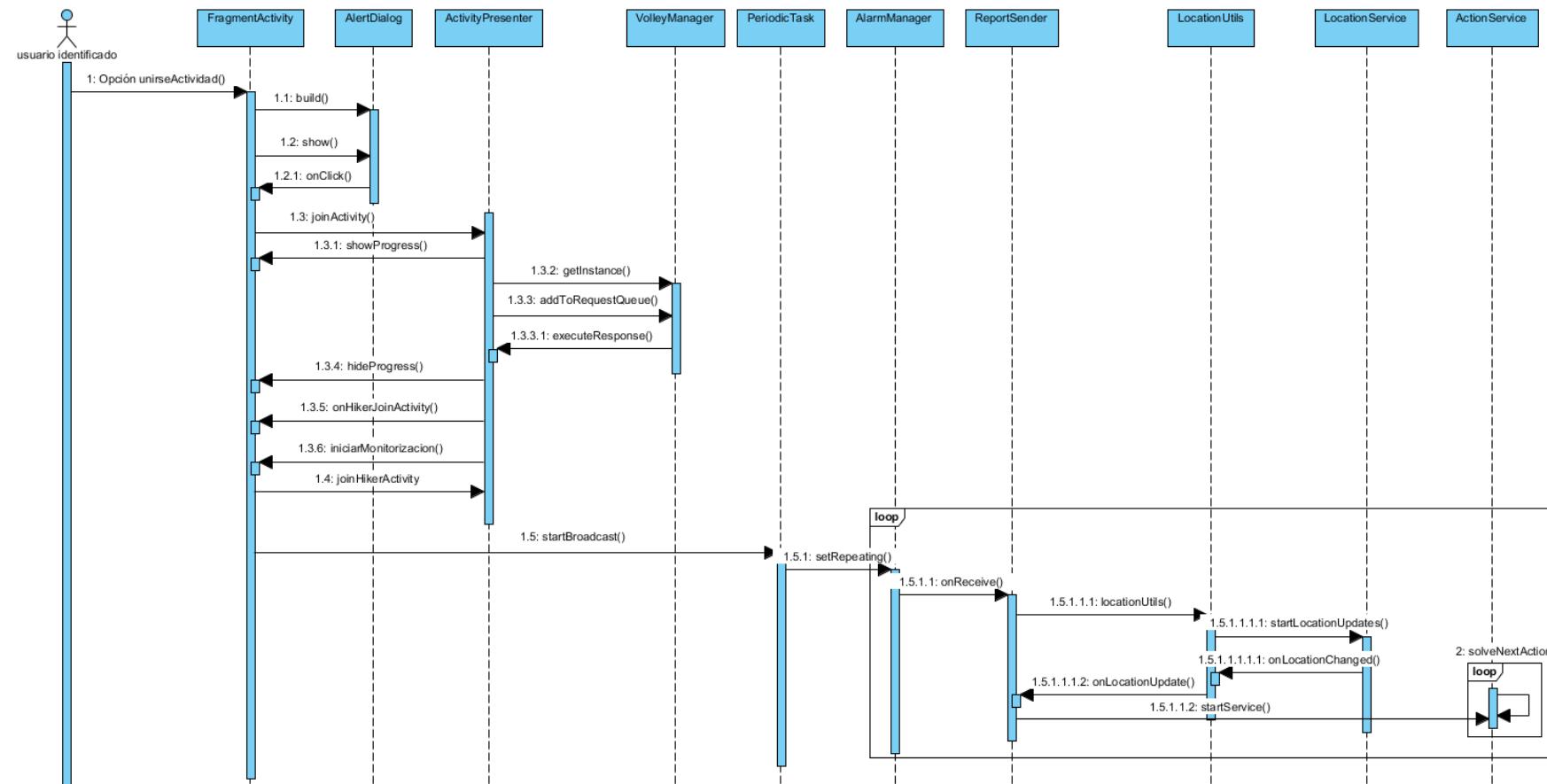


Figura 96 Diagrama de interacción – Unirse actividad

Desde el fragment de una actividad en la aplicación móvil, entre otras acciones, un usuario que no está unido a la actividad puede participar en la misma.

A través de la password de la actividad, el usuario puede unirse a ella y comenzar a ser monitorizado. Además, el dispositivo manda una petición POST al servidor con los datos del hiker que se ha unido a la actividad.

Una vez unido a la actividad el dispositivo de manera automática bajo el tiempo establecido por el propio usuario recoge reportes de posición. Comprueba entonces si tiene alguno pendiente de envío y envía el que acaba de recoger.

5.4.6 Monitorizar actividad

A través de la vista de una actividad, entre otras acciones, el usuario que esté unido a la misma y además esté en curso, podrá monitorizar la actividad.

A través de la monitorización la aplicación muestra al usuario el mapa con el track de la ruta a seguir. Además, muestra una lista de los excursionistas que están unidos en la actividad en ese momento, pudiendo mostrar la posición actual de cada uno de ellos.

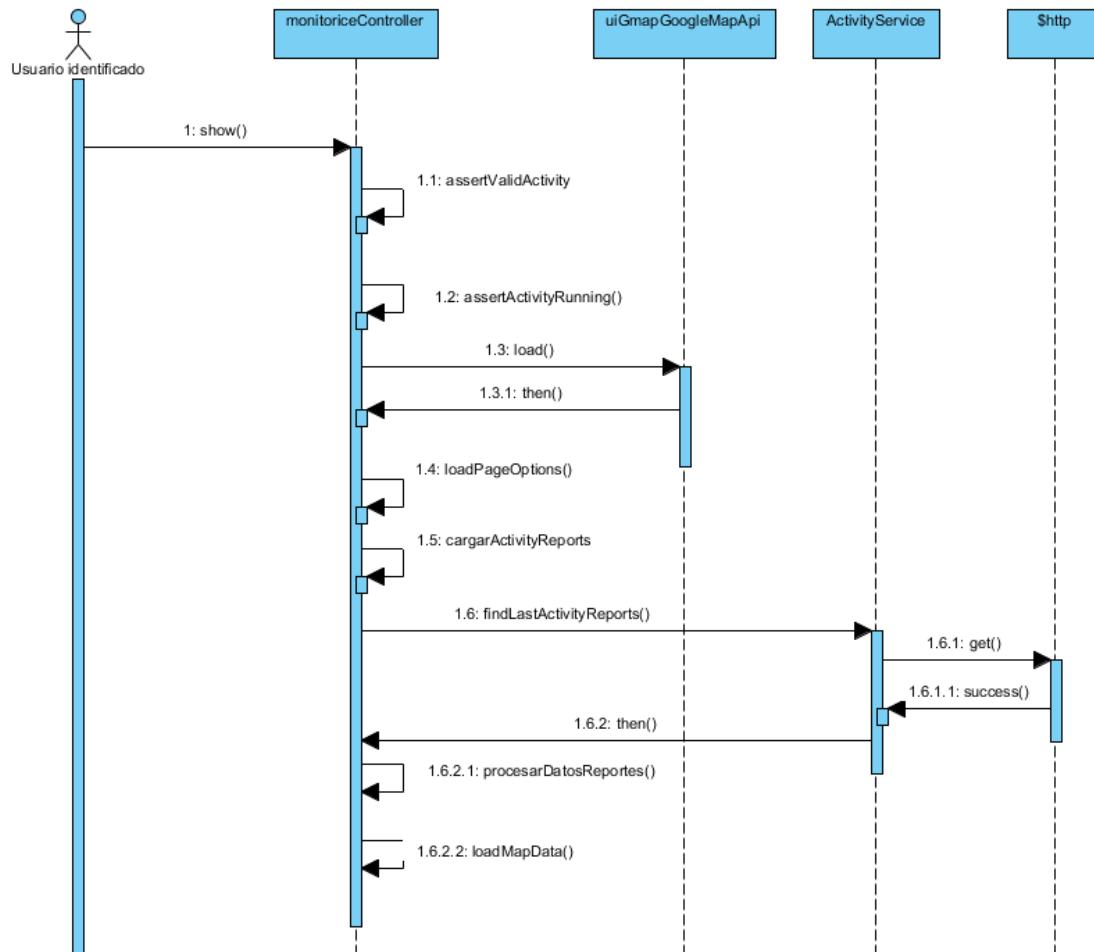


Figura 97 Diagrama de interacción – Monitorizar actividad

5.4.7 Vincular cuenta

Un usuario invitado puede vincular su cuenta en la aplicación móvil a través de la vista del login.

Además, si lo desea, puede de manera opcional marcar la opción para recordar los datos, y de esta manera, en logins sucesivos el dispositivo reconocerá su cuenta sin solicitar las credenciales.

Dichas credenciales cuando son almacenadas la aplicación lo hace en las Settings del dispositivo, implementadas en este caso con SharedPreferences.

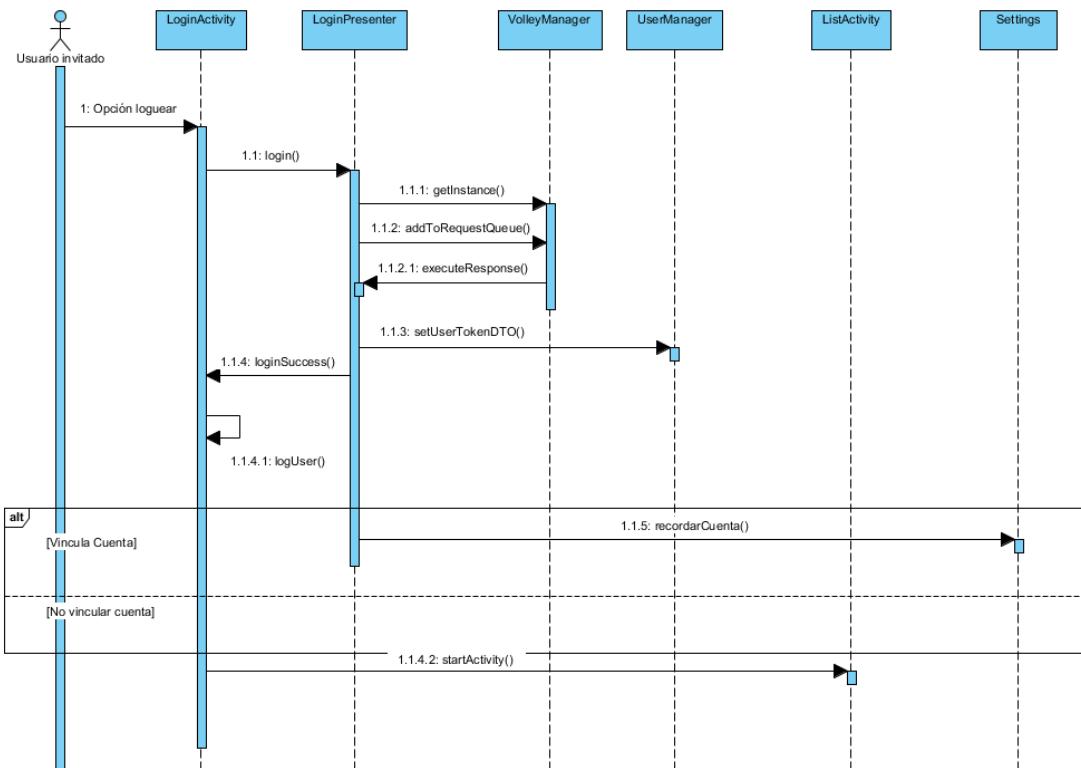


Figura 98 Diagrama de interacción – Vincular cuenta

5.5 Diseño de la base de datos

5.5.1 Descripción del SGDB usado

Tal y como se pudo ver en el diagrama de despliegue, el sistema gestor de base de datos utilizado para la aplicación es PostgreSQL. De la misma manera que se explicó anteriormente, se trata del SGDB más avanzado de código abierto actualmente.

Es orientado a objetos y está mantenido por una gran comunidad de programadores que contribuyen al proyecto versión tras versión.

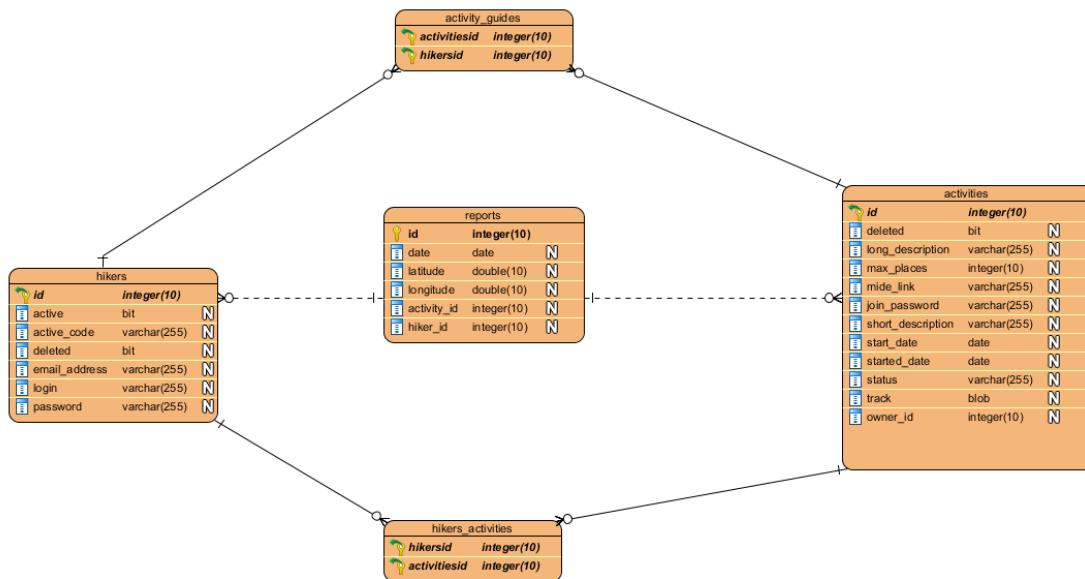


Figura 99 Diagrama Base de datos

5.5.1.1 Hikers

Se inserta una fila en la tabla hiker por cada usuario registrado en el sistema. La clave primaria por la que se identifican es por la id.

El campo active controla que el usuario haya sido activado ya o no después del registro a través del active_code. El campo deleted es el utilizado para el borrado lógico del usuario.

Los campos email_address, login, password, almacenan el correo electrónico, el nombre de usuario y la password (encriptada en MD5) respectivamente.

5.5.1.2 Activities

Se inserta una fila en la tabla activities por cada actividad que se crea en el sistema. Son identificadas por su clave primaria, la id.

Los campos que completan una actividad son: deleted (borrado lógico), long_descripcion y short_descripcion, max_places, mide_link (enlace a descripción de la ruta),

join_password, start_date y started_date, status, track (contenido del track de la ruta, el cual será interpretado por los clientes), y la id del hiker que creó la actividad

5.5.1.3 *Reports*

Se inserta una fila en la tabla reports por cada reporte que es generado en la aplicación. La información que contiene son aquellos puntos GPS formados por la latitud y la longitud.

Además, contienen otra información como la fecha en la que se recogió el reporte, la actividad a la que pertenece y el hiker al que pertenece.

Capítulo 6. Implementación del Sistema

6.1 Estándares y Normas Seguidos

6.1.1 Java Code Conventions

Las Java Code Conventions son un conjunto de normas, pautas y buenas prácticas a seguir estandarizadas para la programación en el lenguaje de programación Java.

Este conjunto de normas ha sido seguido para la programación tanto del servidor como de la aplicación móvil, ambos desarrollados en el lenguaje de programación Java.

Las Java Code Conventions existen debido a que el 80% del tiempo de vida de una pieza de código se emplea en mantenimiento del mismo. Además, con vistas a que otra persona pueda reutilizar el código, habiendo seguido estas normas, la tarea será mucho más sencilla. Pueden ser encontradas en el siguiente enlace: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

6.1.2 Android Code Style and Guidelines

Muy similar a las JCC vistas anteriormente, existen unas normas establecidas por Android de codificación para programadores de esta plataforma.

Estas normas, marcadas como de seguimiento estricto y no como simples sugerencias, incluyen entre otras cosas, tratamiento correcto de excepciones, comentarios, declaración de variables, estilo de código, etc...

Estas normas pueden ser encontradas en el siguiente enlace: <https://source.android.com/source/code-style>

6.1.3 Usabilidad móvil y web: Material Design

Tanto para la aplicación móvil como para la página web se ha tratado de mejorar la experiencia de usuario a través de mejorar la usabilidad de ambas aplicaciones.

En concreto, además, para la página web se han seguido las pautas establecidas por las pautas establecidas por el estilo Material Design.

Además, se han tenido en cuenta siempre una serie de pautas a seguir y respetar:

- Agrupar toda la funcionalidad relacionada en una misma ventana para no hacer al usuario cambiar entre varias.

- Disminuir al mínimo posible el número de pasos a realizar para completar acciones.
- Destacar la información relevante.
- Uso de colores adecuados.
- Tamaño de textos adecuados que garanticen su legibilidad.

6.2 Lenguajes de Programación

Para el desarrollo de la aplicación móvil y el servidor se ha utilizado el lenguaje de programación Java. Para el desarrollo de la página web se ha utilizado Javascript y otros complementarios como HTML, CSS o JSON.

6.2.1 Java

Se trata de un lenguaje de programación de alto nivel, orientado a objetos ejecutado en una máquina virtual JVM independiente de la arquitectura del sistema en la que esté contenida.

Java es uno de los lenguajes de programación más populares y utilizados para desarrollo de aplicaciones cliente-servidor.

La versión de Java utilizada para el desarrollo del proyecto ha sido la 1.8.

6.2.2 Javascript

Utilizado para el desarrollo de la aplicación web, Javascript es un lenguaje de programación interpretado, esto es, no se compila, sino que se interpreta instrucción a instrucción, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Javascript es un lenguaje muy utilizado en aplicaciones del lado cliente pudiendo ejecutarse en navegadores web.

6.3 Herramientas y Programas Usados para el Desarrollo

6.3.1 Spring Tool Suite

Se trata de un entorno de programación para implementar aplicaciones que utilicen el framework Spring. Está desarrollado sobre las últimas versiones de Eclipse por lo que aglutina además toda su funcionalidad.

Spring Tool Suite, además, está pre-configurado con una instalación básica para el despliegue de aplicaciones en servidores tanto de manera local, como virtuales o en la nube.

La versión utilizada de este entorno de desarrollo ha sido la 3.8.3.RELEASE

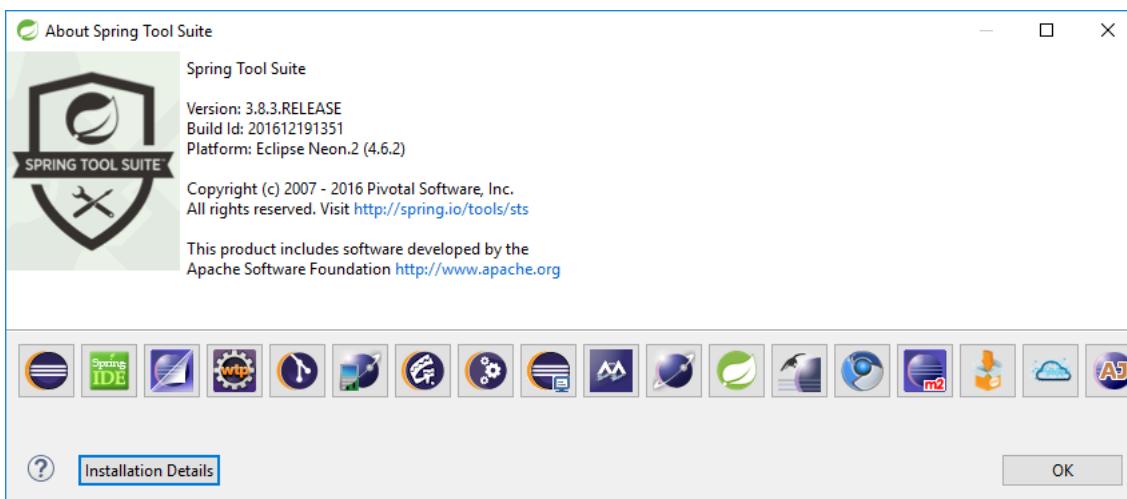


Figura 100 Versión de uso del IDE Spring Tool Suite

6.3.2 Android Studio

Se trata del entorno de desarrollo integrado y oficial para la implementación de aplicaciones en la plataforma Android. Eclipse fue reemplazado por este IDE de manera oficial para el desarrollo de estas aplicaciones a partir de diciembre de 2014.

Al contrario que STS, está basado en IntelliJ Idea JetBrains y actualmente existe para las plataformas Microsoft Windows, macOS y GNU/Linux.

Entre otras muchas características de este entorno de desarrollo se destacan las siguientes utilizadas durante la implementación de la aplicación móvil:

- Renderizado en tiempo real
- Consola para el desarrollador
- Editor de interfaces de usuario
- Virtualización de dispositivos Android en cualquier versión.

La versión utilizada de este IDE ha sido la 2.2.0 pero durante el desarrollo se decidió actualizar a la 2.3.1.



Figura 101 Versión del uso del IDE Android Studio

6.3.3 Atom

Para el desarrollo de la aplicación web se ha utilizado el editor de código fuente Atom. Escrito en Node JS y desarrollado por GitHub, Atom está pensado para la implementación de aplicaciones web. Soporta un gran abanico de lenguajes de programación, entre los que se encuentra Javascript, el utilizado para la aplicación web.

La versión utilizada de este software ha sido la 1.17.2

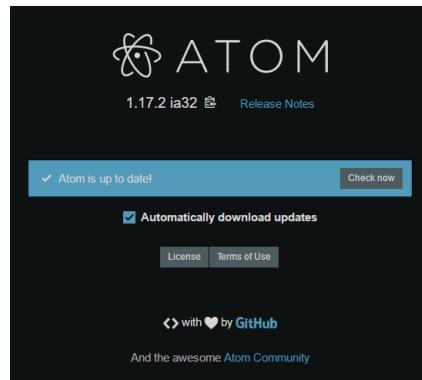


Figura 102 Versión de uso de Atom

6.3.4 PgAdmin 4

Se trata de un entorno de escritorio visual instalable en múltiples plataformas y que permite conectarse a bases de datos PostgreSQL para su gestión y administración de manera gráfica o por SQL.

La versión utilizada ha sido PgAdmin 4 v 1.3

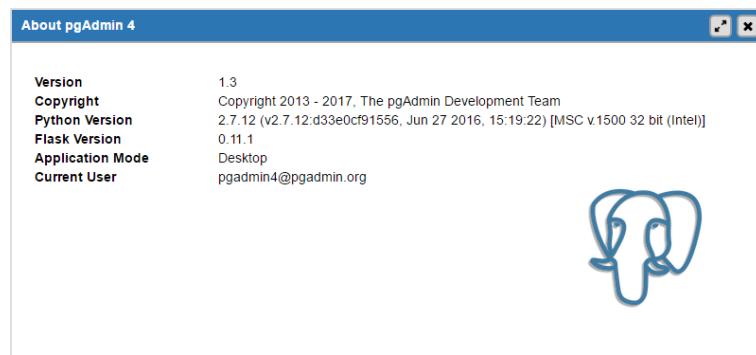


Figura 103 Versión de uso de PgAdmin 4

6.3.5 Visual Paradigm

Se trata de una herramienta para desarrollo UML y modelado de proceso de negocio (BPMN). Tiene funcionalidad para aplicar ingeniería inversa y de código obtener diagramas.

Ha sido utilizado para el modelado de todos los diagramas utilizados para la documentación del proyecto.

La versión utilizada de este software ha sido la 14.1 FREE TRIAL 30 DAYS VERSION al tratarse de un software de pago, a diferencia de los citados anteriormente en este mismo apartado.



Figura 104 Versión de uso herramienta Visual Paradigm

6.3.6 Turbo.net y Browsershots

Ambas herramientas han sido utilizadas para la realización de las pruebas de accesibilidad de la página web. Lo que permiten es ejecutar en casi cualquier navegador web existente, en la versión deseada, la página web que introduzcas.

De esta manera se puede comprobar la compatibilidad de cualquier página web o sistema con los diferentes navegadores y sus correspondientes versiones.

6.4 Librerías usadas para el desarrollo

6.4.1 Angular 1

Framework de Javascript de código abierto utilizado para crear y mantener aplicaciones web de una sola página. Ha sido utilizado como framework principal en el desarrollo de la aplicación web del proyecto.



Figura 105 Logo AngularJS

Está diseñado para implementar aplicaciones web utilizando el patrón Modelo Vista Controlador. AngularJS hace un especial hincapié en las pruebas software y las mejora gracias a que elimina la necesidad de manipular el DOM de manera directa desde la lógica de la aplicación.

6.4.2 Angular Material

Se trata de una librería de componentes implementada bajo las estrictas especificaciones de Material Design de Google. Provee a los programadores de un gran abanico de componentes IU basados en Material Design.



Figura 106 Logo Angular Material

La versión utilizada ha sido la única estable liberada, la 1, que además solo es compatible con la versión 1 de AngularJS. Ha sido utilizado como framework frontend de la aplicación web, con el fin así de cumplir criterios de usabilidad.

6.4.3 Angular Google Maps

Para una mejor compatibilidad de los mapas de Google con el framework Angular 1, se decidió utilizar en la aplicación web una implementación de código abierto para Google Maps, Angular Google Maps.

Desarrollado en CoffeeScript y Javascript, está totalmente integrado con Google Maps en aplicaciones desarrolladas con Angular. Ofrece una gran cantidad de directivas para manejo de mapas, además de acceso directo a los objetos de Google Maps y, por ende, a su propia librería.

6.4.4 OSMDroid

Librería de mapas utilizada en la aplicación móvil. Permite reemplazar el MapView de Android sin perder ninguna funcionalidad. Una de sus principales características, y muy usada en la aplicación móvil es que esta librería permite utilizar proveedores de tiles de mapas externos que son superpuestos mediante capas al mapa de la aplicación.

Además, tiene un sistema de caché de datos transparente al programador y que puede ser configurada, que almacena en el dispositivo las partes de los mapas que ya hayan sido cargadas con anterioridad.

6.4.5 Fabric

Se trata de una herramienta que ayuda al desarrollo de aplicaciones por medio de sus múltiples módulos, entre ellos, el utilizado en el proyecto, crashlytics.

Fabric se configura en la aplicación móvil y gracias al módulo de crashlytics se obtiene, en tiempo real, información de errores que pueda suceder en la aplicación en cualquier usuario, cualquier dispositivo.



Figura 107 Estado de OCAM – móvil en Fabric

Además, la información aportada por Fabric al desarrollador es muy amplia indicando la versión del dispositivo, la traza del error, orientación de pantalla, memoria libre, información de procesos en el momento del crash... etc.

6.4.6 Volley

Se trata de una librería HTTP desarrollada para usarse en aplicaciones móviles desarrolladas en la plataforma Android, que facilita y optimiza las peticiones HTTP.

Entre otras características, las más utilizadas o las que más beneficio han aportado de esta librería son:

- Peticiones múltiples concurrentes
- Priorización de peticiones
- Políticas transparentes de caché de datos

6.4.7 GreenRobot: greenDAO

GreenDAO, perteneciente a GreenRobot es un ORM rápido en ejecución y de uso implementado para Android que mapea objetos a bases de datos SQL Lite internas del dispositivo. Está optimizado para Android y ofrece muy buen funcionamiento consumiendo el mínimo posible de memoria.

Entre otras características aportadas por la librería, ofrece:

- Estable: fue lanzado en 2011 y ha sido utilizado para múltiples y famosas aplicaciones.
- Simple de uso: tiene una API disponible al programador que facilita la familiarización con la librería.
- Ligero: la librería no es pesada ni causa ralentización en las aplicaciones en las que es instalada.
- Encriptación: permite la encriptación de los datos que se almacenan en la base de datos, soportando para ello SQLCipher.
- API para la realización de queries.

6.4.8 Gson

Librería Java para convertir objetos Java en JSON y desde JSON a objeto Java. Además, a diferencia de otras librerías que pretenden el mismo objetivo, Gson soporta objetos complejos que incluyen herencia, relaciones con otros objetos o tipos genéricos.

6.4.9 JavaMail API

Se trata de una librería independiente al framework donde se ejecute para el envío de mensajes. Ha sido utilizada en el servidor para el envío de correos electrónicos durante el proceso de registro en la aplicación.

6.4.10 Jasmine

Jasmine es un framework para la realización de pruebas de comportamiento de aplicaciones desarrolladas en Javascript.



Figura 108 Logo Jasmine

Para la utilización de Jasmine no se necesita ningún otro framework o librería Javascript, ya que no depende de nada más, ni tampoco manipula el DOM durante las pruebas. Para la realización de los tests se usa código Javascript con una sintaxis de funciones de muy alto nivel que facilitan la elaboración de los mismos.

6.4.11 JUnit

Se trata de un framework para escribir tests. Se utiliza para tests unitarios de programas escritos en Java. Incluye mecanismos gráficos para ver los resultados de los tests que se ejecutan.



Figura 109 Logo JUnit

Dos de las herramientas de desarrollo más conocidas, Eclipse y NetBeans, cuentan con plug-ins para el desarrollo y ejecución de pruebas unitarias con JUnit.

6.5 Problemas Encontrados

En su mayoría, durante el desarrollo de la aplicación Android surgieron algunos problemas que fueron solventados, en su mayoría, mediante la introducción de algunas de las librerías citadas anteriormente.

6.5.1 Consumo de batería

En un primer momento no se hizo especial hincapié en que los consumos de batería ya que la principal preocupación era poder satisfacer todos los requisitos funcionales. Una vez se cumplió esto y se llegó a tener la aplicación funcional completa, nos dimos cuenta a través de pruebas que los consumos de batería no eran los suficientemente optimizados en comparación a otras aplicaciones del mercado, mencionadas algunas en el análisis de alternativas.

Los dos puntos clave de consumo de batería de la aplicación son la ejecución, mediante el Broadcast Receiver, de tareas en segundo plano y la activación del GPS de manera constante para requerir posicionamiento al dispositivo.

Se analizaron estos dos requisitos y la manera de mejorarlos para, sin alterar la funcionalidad, bajar los consumos de batería. Para esto se decidió cambiar la gestión que se hacía del GPS: en las primeras versiones estaba encendido de continuo, ahora solamente se enciende, o, dicho de otra manera, se conecta la API, cuando realmente se va a solicitar una actualización de posición. Esto baja notablemente el tiempo que el GPS está encendido, incluso cuando la frecuencia de monitorización especificada por el usuario es la mínima (un minuto).



Figura 110 Consumo de batería aplicación móvil

Tal y como se aprecia en la figura anterior, se realizó un consumo muy similar a otras aplicaciones que realizan tareas similares durante el desarrollo de una actividad al aire libre. Cabe destacar que la realización de esta comparación no es con el fin de valorar la validez de

una aplicación sobre otra sino para establecer una referencia en consumos de batería bajo condiciones de trabajo similares.

En el caso de OCAM, con la nueva solución de tratamiento de GPS, se consigue reducir el consumo de batería, el cual anteriormente se situaba por encima del 11% en períodos de funcionamiento similares al de la figura.



Figura 111 Desglose de consumo de batería
OCAM



Figura 112 Desglose consumo de batería
aplicación análoga

En la figura anterior queda claramente ilustrado el funcionamiento explicado. La actividad en la que fue monitorizado el dispositivo duró aproximadamente dos horas. Tal y como se aprecia en la fotografía que ilustra la explicación anterior, el GPS ha estado conectado durante todo el desarrollo de la actividad, pero solamente activo durante 1 minuto y 38 segundos. Este es el tiempo que ha sido necesario para obtener la posición del dispositivo teniendo la frecuencia de monitorización al máximo, es decir, 1 minuto entre reporte y reporte.

Finalmente, se puede ver como el consumo de batería reportado por la aplicación es de 278mAh mientras que en el caso de su análoga que funcionó el mismo tiempo fue de 331, teniendo incluso, en el caso de OCAM, un mayor tráfico de datos en red y otra estrategia para obtener la posición, si bien es cierto que la aplicación análoga finalmente reportó un mayor tiempo de uso de GPS.

Finalmente, una vez llegados a esta solución al problema del consumo de batería, surgió la pregunta de si esto sería realmente eficiente en el sentido de obtener posiciones GPS sin aumentar el margen de error. Para ello, se realizó la misma comprobación que en el caso anterior, es decir, bajo el mismo funcionamiento en una aplicación análoga comparar las dos trayectorias del dispositivo.

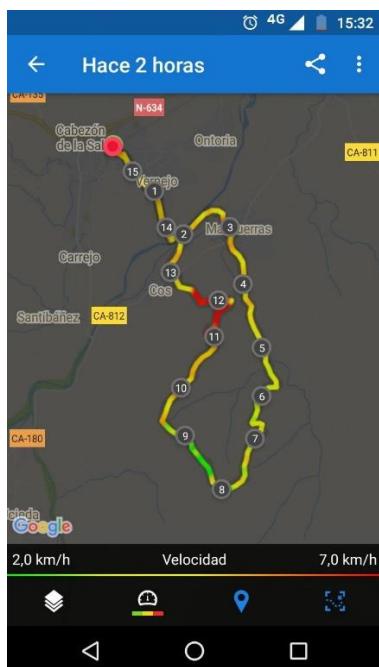


Figura 113 Resumen de actividad sobre el mapa en actividad análoga

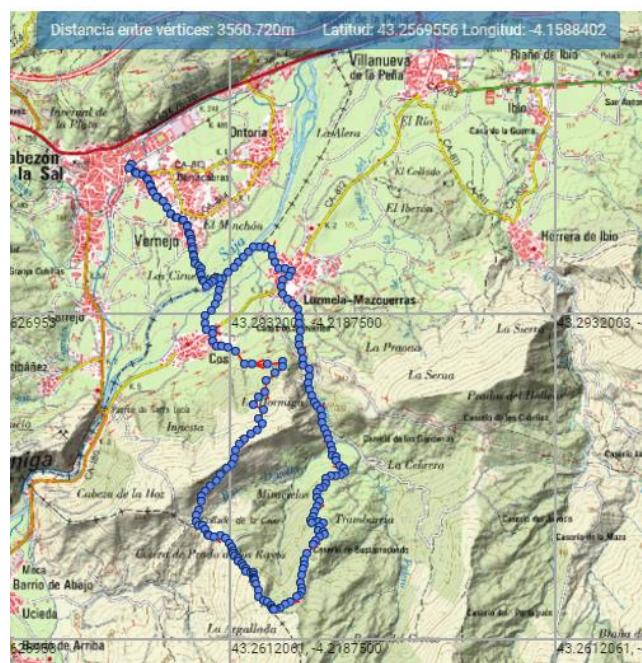


Figura 114 Resumen de actividad en OCAM

Tal y como se aprecia en las anteriores, la precisión obtenida en la aplicación análoga y en OCAM es muy similar, obteniéndose resultados de la actividad muy similares.

6.5.2 Precisión de los reportes

Para llegar a la precisión que se mostró en imágenes anteriores se realizó un cambio una vez acabada la funcionalidad de la aplicación que significó una disminución del error en posiciones GPS.

En primera instancia el tipo de dato que almacenaba las posiciones GPS era float. Esto, con el tiempo, comparando las posiciones, decimal a decimal, que obtenía el dispositivo móvil, con las que luego llegaban al servidor, me dí cuenta de que se perdían decimales para algunas posiciones en determinados reportes de posición.

Tipo	Precisión	Tamaño
float	7 dígitos	32 bits
double	15-16 dígitos	64 bits

Tabla 16 Comparativa float – double Java

Tal y como se ilustra en la tabla anterior el tipo float tiene una precisión de 7 dígitos por lo que se podrían estar perdiendo por reporte hasta 9 dígitos. Esto, como se comentó antes, no significó un cambio del que realmente se pudieran obtener grandes resultados, pero sí que mejoró la precisión de cada reporte obtenido.

6.5.3 Requerimiento de permisos

Uno de los problemas con los que ha habido que lidiar durante el desarrollo de la aplicación Android son los permisos. Android es un sistema operativo en el que cada aplicación se ejecuta con una identidad propia, y se basa en el uso de permisos por cada aplicación.

Cada una de las aplicaciones aporta en tiempo de instalación los permisos necesarios que luego utilizará en ejecución. Sin embargo, desde la versión 6.0 de Android esto ha dejado de ser así.

Dependiendo del dispositivo móvil, los mapas y la localización obtenida en la aplicación realizaban un correcto funcionamiento o no en base precisamente a la versión de Android del propio dispositivo. La solución a esto fue, según lo que marca la documentación para la versión 6.0 de Android, pedir los permisos al usuario siempre en tiempo de ejecución y bloquear aquellas funcionalidades de la aplicación para las cuales el usuario no haya dado permiso explícito.

6.5.4 Almacenamiento de datos para funcionamiento offline

Una de las principales funcionalidades de la aplicación móvil es su funcionamiento offline. Tal y como se describió en el capítulo de análisis del proyecto, la aplicación es capaz de almacenar acciones y datos para volcarlos en cuanto recobrarse la conexión.

La primera aproximación a esta funcionalidad fue mediante el uso de bases de datos SQLite, interna del dispositivo. Utilizando esto se llegó a la conclusión que no era una solución viable ya que ralentizaba el funcionamiento de la aplicación y la implementación era muy costosa.

Para ello, la solución fue la inclusión en la aplicación de la librería greenDAO descrita anteriormente.

6.5.5 Ejecución de tareas en segundo plano

Uno de los problemas más destacables acaecido durante la implementación de la aplicación móvil y para el cual se desarrollaron varias soluciones, fue el de las tareas en segundo plano.

Para que la aplicación móvil pueda funcionar correctamente, tal y como se describió en la fase de análisis, debe realizar tareas de manera periódica (obtener y mandar la localización, entre otras) en segundo plano. Para llevar a cabo esto, gracias a la API de Android, existe más de un método, cada uno con sus propias características, ventajas y desventajas.

Cabe destacar que es un requisito el que la aplicación consuma el mínimo de batería posible y el uso de datos móviles esté optimizado también para no exceder las cuotas de uso.

Con el fin de garantizar esto se implementaron varias opciones hasta llegar a la opción más adecuada y actual:

1. Se implementó la tarea en background mediante un servicio Android. Esta manera de llevar a cabo las tareas en segundo plano era la más óptima en cuanto a rendimiento, no consumía a penas batería. El problema de esta opción, y por la que se tuvo que descartar, es que Android elimina el proceso cuando la aplicación es eliminada por el usuario, o incluso, dependiendo el caso, al bloquear el dispositivo.
2. Implementarlo mediante un broadcast receiver. Menos eficiente en cuestiones de batería que la posible solución anterior, suple todos los problemas de ejecución en segundo plano. No garantiza una ejecución exacta en el periodo de tiempo que se le indique, pero nunca va a ser eliminado por el sistema operativo a menos que el usuario lo haga de manera explícita.

Por otro lado, el broadcast receiver presentaba un problema y es que el sistema operativo, para garantizar la seguridad del dispositivo, corta el flujo de ejecución aproximadamente diez segundos tras comenzar o bien una vez ejecutada la última línea de código. Esto era un problema para el envío de la posición del excursionista o bien para la ejecución de tareas almacenadas offline debido a que todo ello se realiza de manera asíncrona.

Para suplir este problema, lo que se hizo fue combinar las dos soluciones. Esto es, una vez se ejecuta el broadcast receiver, bajo la periodicidad indicada por el usuario, éste lanza un servicio Android desde el cual es enviada/s la petición/es HTTP para completar todas las acciones pendientes. Hasta que no se recibe la respuesta 200 OK del servidor a dichas peticiones no se eliminan de la memoria interna del dispositivo. De esta manera y mediante esta implementación del problema, se garantiza que nunca se van a perder datos o acciones.

6.6 Descripción detallada del sistema desarrollado

6.6.1 Diagrama de paquetes servidor

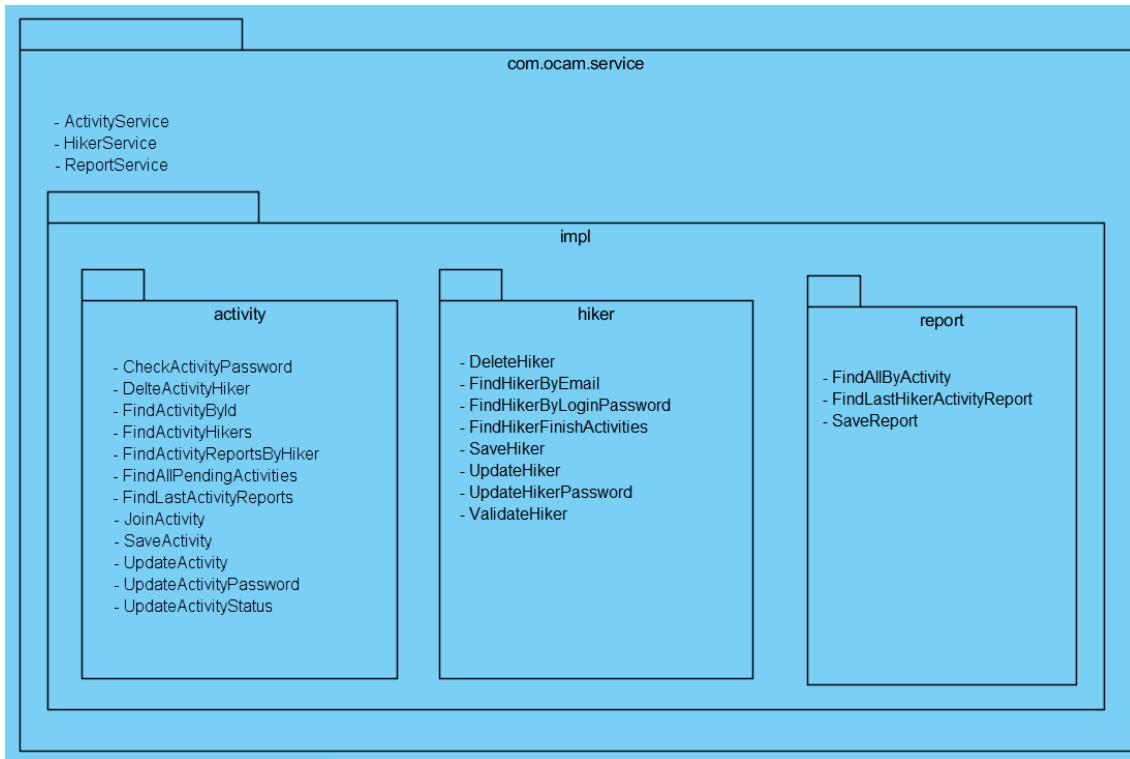


Figura 115 Diagrama de paquetes servidor com.ocam.service

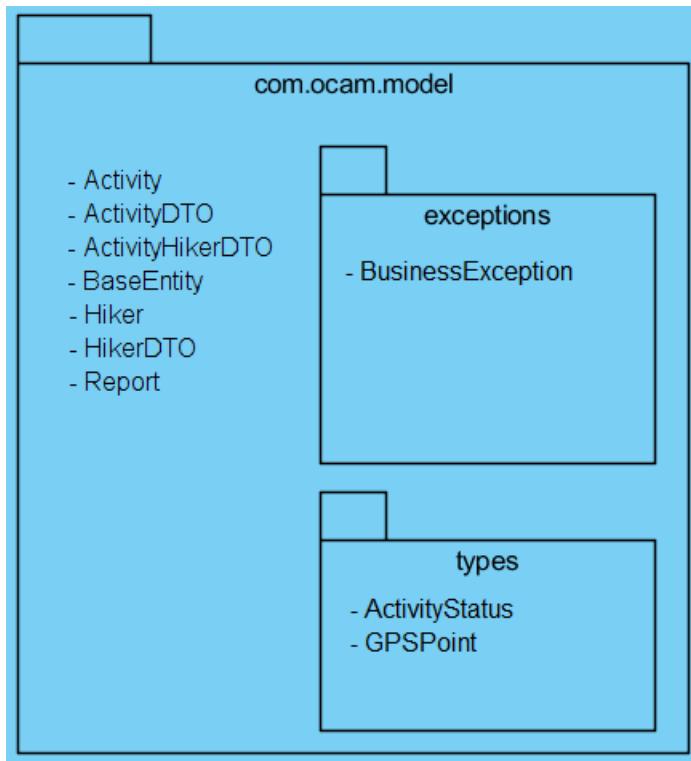


Figura 116 Diagrama de paquetes servidor com.ocam.model

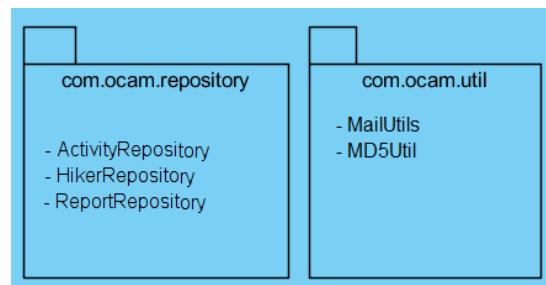


Figura 117 Diagrama de paquetes servidor com.ocam.util, com.ocam.repository

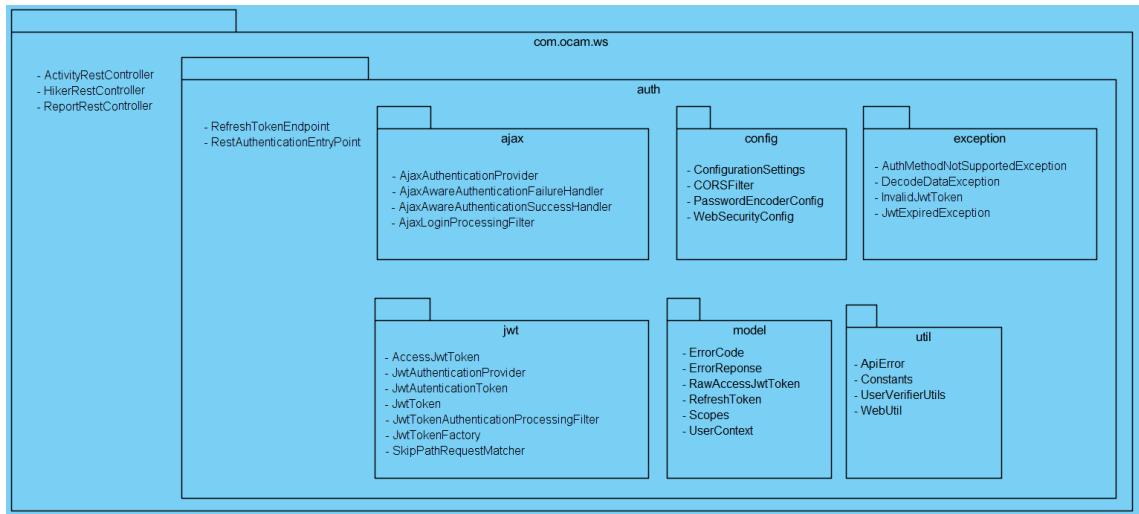


Figura 118 Diagrama de paquetes servidor com.ocam.ws

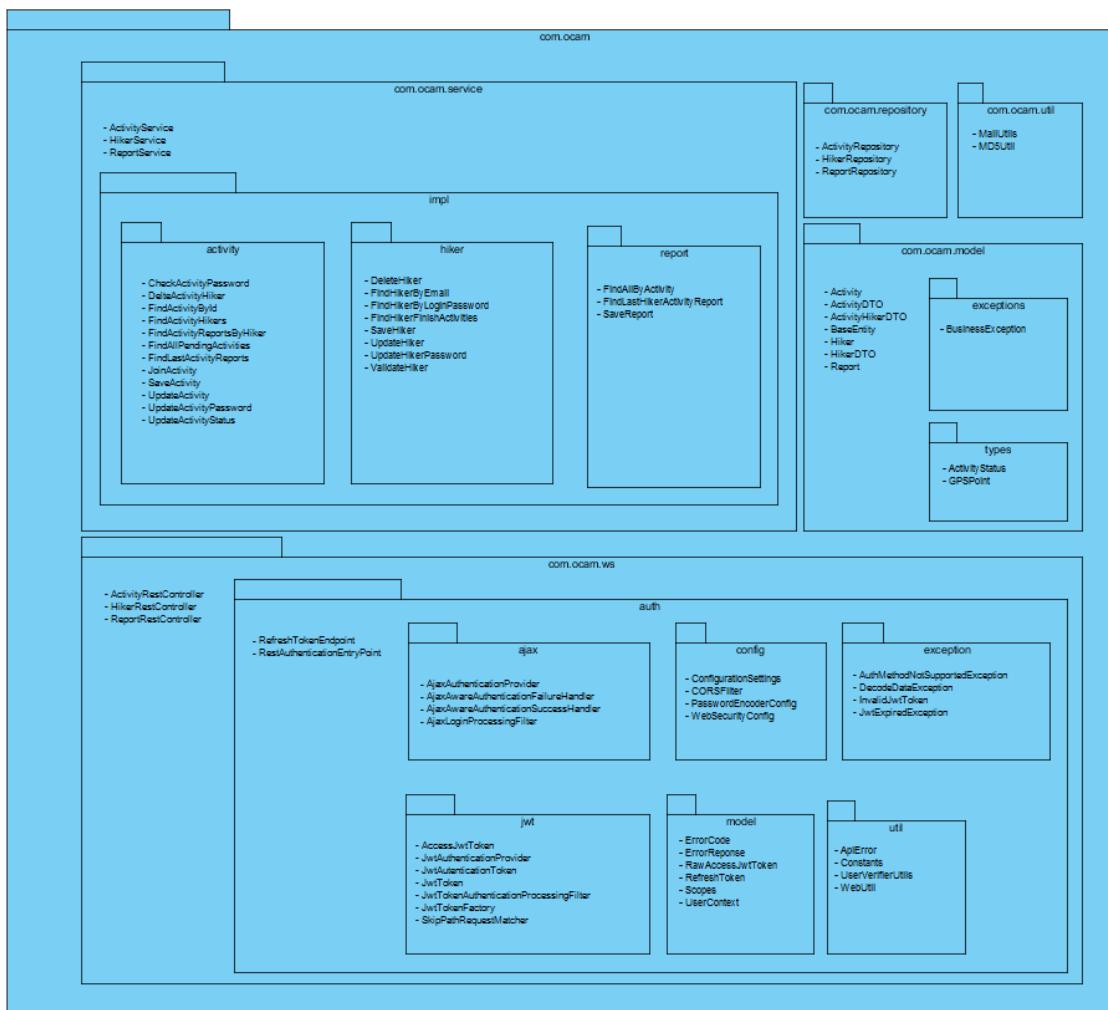


Figura 119 Diagrama de paquetes servidor

6.6.2 Diagrama de paquetes aplicación móvil

La aplicación móvil, tal y como se describió en el capítulo de diseño se ha realizado usando el patrón modelo vista presentador. Las clases han sido agrupadas en paquetes incluyendo aquellas que sean susceptibles de ser modificadas si otras también lo son en el mismo paquete.

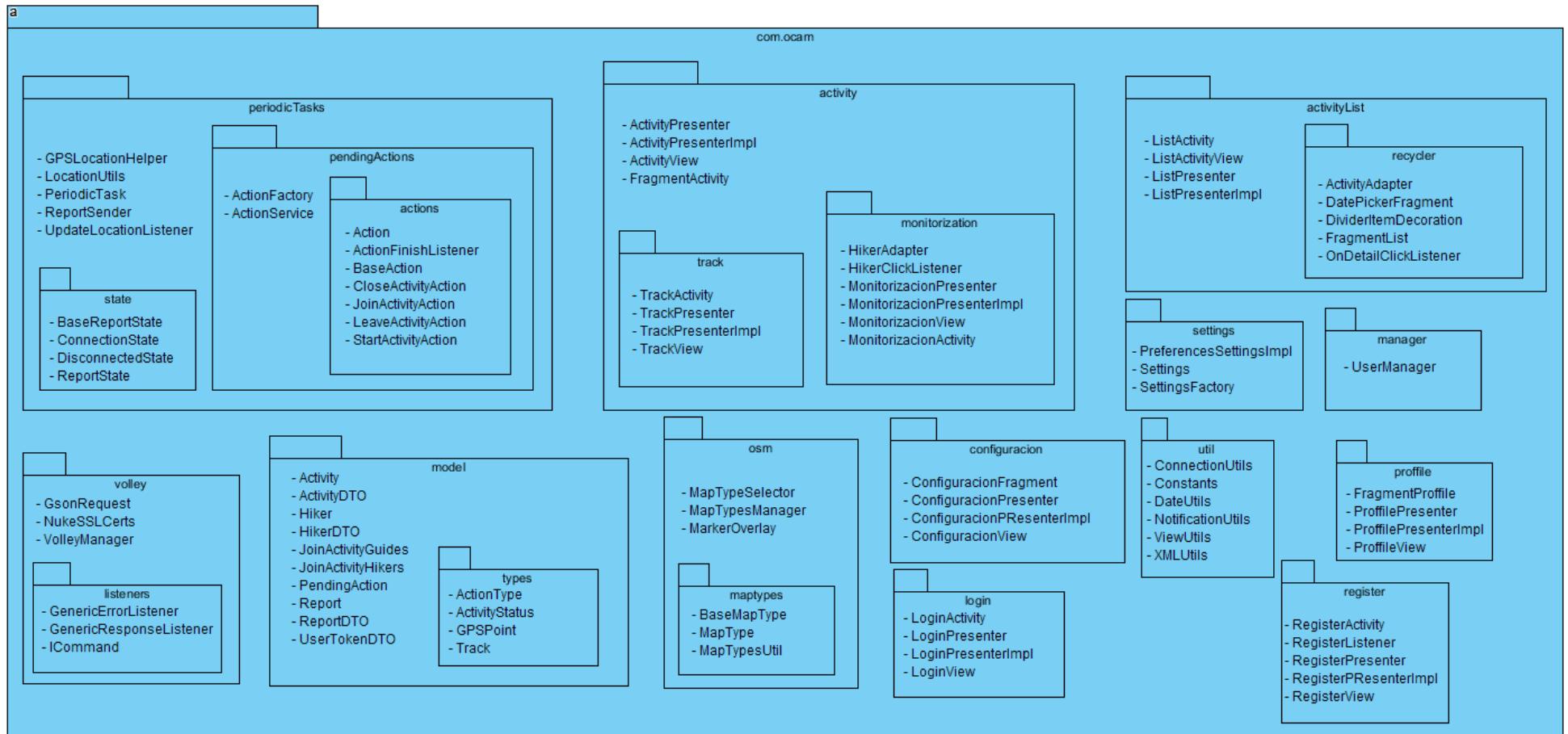


Figura 120 Diagrama de paquetes aplicación móvil

6.6.3 Descripción elementos aplicación web

Tal y como se describió en el capítulo de diseño, la aplicación web sigue el patrón Model View View Model, implementada con AngularJS 1. En este apartado se van a describir los principales elementos que componen la aplicación, así como la organización de la misma con el fin de describir de una manera generalizada la manera en la que ha sido desarrollada.

La aplicación ha sido dividida en cuatro módulos muy diferenciados entre ellos, que se describen a continuación:

- Ocam: módulo principal que importa el resto de módulos y realiza la configuración más básica de la aplicación
- Public: módulo que gestiona todos los controladores y componentes de la parte pública de la aplicación: acceso, registro, listado de actividades... etc.
- Private: módulo que gestiona todos los controladores y componentes de la parte privada de la aplicación: crear actividad, perfil, monitorizar, detalle de actividad, listado... etc.
- Common: módulo que contiene elementos comunes a public y private, como constantes, internacionalización, lista de actividades, header, autenticación... etc.

A continuación, se explican los principales elementos que se pueden encontrar en cada uno de los módulos descritos.

6.6.3.1 Rutas

Los ficheros de rutas de la aplicación móvil siguen el nombre *modulo.routes.js*, siendo el módulo el público o el privado. Se trata de la configuración de los estados y las vistas por las que el usuario puede pasar.

```
(function() {
  'use strict';

  angular.module('public')
  .config(routeConfig);

  /**
   * Public routing of the application
   */
  routeConfig.$inject = ['$stateProvider', '$locationProvider'];
  function routeConfig ($stateProvider, $locationProvider) {
    $stateProvider
      .state('public', {
        abstract: true,
        templateUrl: 'src/common/header/header.html'
      })
      .state('public.actList', {
        url: '/',
        controller: 'activityListController',
        controllerAs: 'actListCtrl',
        templateUrl : 'src/common/actList/actList.html',
        data: {
          authorization: false
        },
        //Inyecta en el controlador las primeras actividades y el número total
        resolve: {
          'list': ['$activityService', function($activityService) {
            return $activityService.findAllPending(0, 10);
          }]
        }
      })
  }
})()
```

```

        'numEle': ['activityService', function(activityService) {
            return activityService.findCountAll();
        }]
    })
.state('public.access', {
    url: '/access/:code',
    controller: 'accessController',
    controllerAs: 'accessCtrl',
    templateUrl : 'src/public/access/access.html',
    data: {
        authorization: false
    },
    params: {
        showError: null
    }
});
})
());
}
)()();

```

En la función *routeConfig* se realiza la configuración de las rutas. En este caso tenemos las del módulo *public*. Se cuenta con un estado *public* abstracto y el resto de estados, hijos de *public* comparten e inyectan a continuación de la plantilla del padre la suya propia. Esto permite insertar la cabecera de la aplicación en cada uno de los estados sin necesidad de repetir código.

Para cada estado se define el que será su controlador, la url, la plantilla html que seguirá, si es un estado que requiera autorización o no, parámetros o incluso si es necesario cargar datos antes de iniciarla.

6.6.3.2 *Controladores*

Un controlador es una clase Javascript que lleva la lógica asociada a una vista de la aplicación. En este caso se va a explicar a partir de *access*. Este controlador gestiona la vista de acceso a la aplicación, la cual tiene a su vez dos componentes, registro y login, que serán explicados más adelante.

Por tanto, la única funcionalidad que en este caso gestiona es la de mostrar la ventana emergente a la hora de validar una cuenta.

```

(function () {
'use strict';

angular.module('public')
.controller('accessController', accessController);

accessController.$inject = ['$stateParams', '$mdDialog', '$translate',
'$translatePartialLoader'];
function accessController($stateParams, $mdDialog, $translate,
$translatePartialLoader) {

    var $ctrl = this;
    var code = $stateParams.code;
    $ctrl.showError = $stateParams.showError;
    $translatePartialLoader.addPart('access');
    $translatePartialLoader.addPart('errors');
    $translate.refresh();

    var showDialog = function() {
        $mdDialog.show({
            controller: 'AccountController',
            controllerAs: 'AccountCtrl',
            templateUrl: 'src/public/access/account/accountdialog.template.html',
            parent: angular.element(document.body),

```

```

        clickOutsideToClose: false,
        locals : {
          'code' : code
        }
      });
    }

    if (code) {
      $ctrl.loading = true;
      showDialog();
    }
  }
})();

```

El controlador, *accessController* recibe inyectado diferentes servicios para gestionar los parámetros de entrada a la vista, poder mostrar ventanas emergentes, importar tablas de traducción y traducir textos de manera dinámica desde código, respectivamente.

Dentro de la función de un controlador y a través del objeto *this* puedes realizar cualquier comunicación bidireccional con la vista, ya sea para leer un dato o para mostrarlo.

6.6.3.3 Componentes

Si una vista tiene diferentes secciones con funcionalidades separadas, con el fin de mejorar la escalabilidad de la aplicación, no repetir código, que éste sea mucho más limpio y legible, se utilizan componentes.

Explicado a alto nivel, un componente es la forma de subdividir un controlador en varios. Además, un componente puede ser inyectado en cualquier parte de la aplicación sin necesidad de repetir código. Simplemente dándole un nombre se puede insertar en una vista como si se tratase de una etiqueta HTML normal:

<nombre_del_componente></nombre_del_componente>

En este caso se va a explicar el componente *loading*. Este componente se utiliza en todas las vistas de la aplicación que impliquen peticiones HTTP al servidor, es decir, aproximadamente la totalidad.

```

(function() {
  "use strict";

  angular.module('common')
  .component('loading', {
    template: '<md-progress-linear ng-if="$ctrl.show" md-mode="indeterminate"/>',
    controller: LoadingController
  });

  LoadingController.$inject = ['$rootScope'];
  function LoadingController ($rootScope) {
    var $ctrl = this;
    var listener;

    $ctrl.$onInit = function() {
      $ctrl.show = false;
      listener = $rootScope.$on('spinner:activate', onSpinnerActivate);
    };

    $ctrl.$onDestroy = function() {
      listener();
    };

    function onSpinnerActivate(event, data) {

```

```

        $ctrl.show = data.on;
    }
}

})();

```

La primera parte de un componente es la definición del mismo. Se añade a un módulo, en este caso, al usarse en parte pública y privada se adjunta al módulo común. El template es el código HTML que tendrá el componente en la vista. Se puede insertar el código en el propio fichero Javascript, o bien indicar la ruta a un fichero HTML. A continuación, se le indica un controlador.

Un componente además puede tener otros parámetros como por ejemplo métodos de comunicación con el controlador de la vista en el que estén insertados, de tal manera que se puedan comunicar con éste e incluso con otros componentes.

En este caso el componente lleva su propio controlador que gestiona la vista, definido de igual manera que vimos anteriormente.

6.6.3.4 *Servicios*

Los servicios de Angular son objetos a los que se pueden inyectar otros servicios y que sirven para ser inyectados en otros controladores con el fin de realizar comunicaciones entre ellos o refactorizar código.

Los servicios en Angular pueden ser de dos tipos, a elección del programador dependiendo de las necesidades:

- Lazy instantiated: se instancian siempre que vayan a ser ejecutados, nunca antes.
- Singleton: se instancian una vez y cada componente que necesite hacer uso de ellos accederá a la misma instancia compartida. Estos son los Factory.

Para explicar los servicios vamos a utilizar, relacionado con el componente anterior, la factoría que intercepta peticiones HTTP, *loading.interceptor.js*.

```

(function() {
"use strict";

angular.module('common')
.factory('loadingHttpInterceptor', LoadingHttpInterceptor);

/**
 * Lanza un evento cada vez que una petición http empieza
 * y acaba.
 */
LoadingHttpInterceptor.$inject = ['$rootScope', '$q'];
function LoadingHttpInterceptor($rootScope, $q) {

    var loadingCount = 0;
    var loadingEventName = 'spinner:activate';

    return {
        request: function (config) {
            if (++loadingCount === 1) {
                $rootScope.$broadcast(loadingEventName, {on: true});
            }
        },
        return config;
    },
}

```

```

        response: function (response) {
            if (--loadingCount === 0) {
                $rootScope.$broadcast(loadingEventName, {on: false});
            }
            return response;
        },
        responseError: function (response) {
            if (--loadingCount === 0) {
                $rootScope.$broadcast(loadingEventName, {on: false});
            }
            return $q.reject(response);
        }
    };
}
})();

```

Al ser una factoría, y tal y como se explicó antes, solo es instanciada una vez. La estructura es muy similar a la de un componente, se define indicando el nombre y la función que llevará la lógica de negocio asociada.

En este caso en la lógica tenemos la definición de un objeto que se suscribe a todas las peticiones HTTP que realiza la página web. Con cada petición lanza un evento que posteriormente es capturado por todos los componentes loading de la vista mostrándolos el tiempo que dure la petición.

6.6.4 Descripción detallada de las clases

En este apartado se van a describir aquellas clases que se consideran más relevantes de la implementación del servidor y la aplicación móvil.

6.6.4.1 Servidor

6.6.4.1.1 ActivityRestController

Clase controladora de la API REST para la entidad Activity. Contiene todos los métodos HTTP POST, GET, PUT y DELETE para esta entidad.

Retorno del método	Método y descripción
<code>org.springframework.http.ResponseEntity<?></code>	<code>allPendingActivities()</code> Devuelve las actividades pendientes de realización o bien que estén en curso
<code>org.springframework.http.ResponseEntity<?></code>	<code>checkActivityPassword(java.lang.Long id, java.lang.String password)</code> Comprueba la password de una actividad
<code>org.springframework.http.ResponseEntity<?></code>	<code>closeActivity(java.lang.Long activityId)</code> Cierra una actividad pasándola a estado CLOSED
<code>org.springframework.http.ResponseEntity<java.lang.Object></code>	<code>countPendingActivities()</code> Retorna el número total de actividades pendientes
<code>org.springframework.http.ResponseEntity<?></code>	<code>find(java.lang.Long id)</code> Devuelve una actividad buscada por ID
<code>org.springframework.http.ResponseEntity<?></code>	<code>findActivityHikerReports(java.lang.Long activityId,</code>

	<code>java.lang.String hikerEmail)</code> Método que busca los reports de un hiker dado para una actividad dada
<code>org.springframework.http.ResponseEntity<?></code>	<code>findActivityHikers(java.lang.Long activityId)</code> Retorna todos los Hiker asociados a una actividad, sean guías o no
<code>org.springframework.http.ResponseEntity<?></code>	<code>findLastActivityReports(java.lang.Long id)</code> Devuelve los últimos reportes de cada hiker de una actividad
<code>org.springframework.http.ResponseEntity<?></code>	<code>joinActivity(java.lang.Long activityId, java.lang.String login, java.lang.String password)</code> Incluye a un hiker como participante de una actividad
<code>org.springframework.http.ResponseEntity<?></code>	<code>leaveActivity(java.lang.Long activityId, java.lang.String login)</code> Elimina un hiker de una actividad.
<code>org.springframework.http.ResponseEntity<?></code>	<code>pendingActivities(java.lang.Integer minResults, java.lang.Integer maxResults)</code> Devuelve las actividades pendientes encontradas entre los rangos indicados por parámetro
<code>org.springframework.http.ResponseEntity<java.lang.Object></code>	<code>save(javax.servlet.http.HttpServletRequest request, ActivityHikerDTO activityHiker)</code> Guarda y actualiza una actividad
<code>org.springframework.http.ResponseEntity<?></code>	<code>startActivity(javax.servlet.http.HttpServletHttpservletRequest request, Activity activityDTO)</code> Da una actividad por comenzada cambiándole el estado a RUNNING
<code>org.springframework.http.ResponseEntity<?></code>	<code>updateActivityPassword(javax.servlet.http.HttpServletHttpRequest request, java.lang.Long activityId, java.lang.String password)</code> Actualiza la password de una actividad

6.6.4.1.2 HikerRestController

Clase controladora de la API REST para la entidad Hiker. Contiene todos los métodos HTTP GET, POST, PUT, DELETE para esta entidad.

Retorno del método	Método y descripción
<code>org.springframework.http.ResponseEntity<?></code>	<code>deleteHiker(javax.servlet.http.HttpServletRequest request, java.lang.String login)</code> Elimina un hiker (borrado lógico)
<code>org.springframework.http.ResponseEntity<Hiker></code>	<code>findHiker(java.lang.String username)</code> Retorna el objeto Hiker buscándolo por el login
<code>org.springframework.http.ResponseEntity<?></code>	<code>findHikerByEmail(java.lang.String email)</code> Comprueba que un email existe en la aplicación
<code>org.springframework.http.ResponseEntity<?></code>	<code>findHikerByLogin(java.lang.String login)</code> Comprueba que un nombre de usuario existe en la aplicación
<code>org.springframework.http.ResponseEntity<?></code>	<code>findHikerFinishActivities(java.lang.S</code>

<code>sposeEntity<?></code>	<code>tring login)</code> Devuelve la lista de actividades en las que participó el hiker y están cerradas
<code>org.springframework.http.ResposeEntity<?></code>	<code>resetPassword(HikerDTO hiker)</code> Genera una nueva contraseña para el hiker
<code>org.springframework.http.ResposeEntity<java.lang.Object></code>	<code>saveHiker(HikerDTO hiker,</code> <code>org.springframework.web.util.UriComponentsBuilder ucBuilder)</code> Guarda un nuevo Hiker en el sistema
<code>org.springframework.http.ResposeEntity<?></code>	<code>validateHiker(HikerDTO hiker)</code> Retorna el objeto Hiker buscándolo por el login
<code>org.springframework.http.ResposeEntity<java.lang.Object></code>	<code>verifyPassword(javax.servlet.http.HttpServletRequest request, HikerDTO hiker)</code> Método para cambiar la password de un hiker

6.6.4.1.3 ReportRestController

Clase controladora de la API REST para la entidad Report. Contiene todos los métodos HTTP GET, POST, PUT, DELETE para esta entidad.

Retorno del método	Método y descripción
<code>org.springframework.http.ResponeEntity<?></code>	<code>findAllByActivity(java.lang.Long id)</code> Devuelve todos los reportes de una actividad
<code>org.springframework.http.ResponeEntity<?></code>	<code>findLastHikerActivityReport(java.lang.Long id, java.lang.String login)</code> Devuelve el último reporte de un hiker de una actividad
<code>org.springframework.http.ResponeEntity<?></code>	<code>save(Report report)</code> Retorna el objeto Hiker buscándolo por el login

6.6.4.1.4 CORSFilter

Se trata de una clase de configuración realizada programáticamente de la seguridad CORS de la aplicación.

Es un filtro HTTP ejecutado antes que los filtros de autenticación y que no discrimina ninguna ruta. Implementa por tanto la clase filter y el método doFilter:

```
@Override
public void doFilter(ServletRequest req, ServletResponse resp,
                      FilterChain chain) throws IOException, ServletException {

    HttpServletResponse response = (HttpServletResponse) resp;
    HttpServletRequest request = (HttpServletRequest) req;

    response.setHeader("Access-Control-Allow-Origin", "*");
    response.setHeader("Access-Control-Allow-Methods",
                      "POST, GET, OPTIONS, DELETE");
    response.setHeader("Access-Control-Max-Age", "3600");

    String path = request.getRequestURI()
                  .substring(request.getContextPath().length());
    if (Constants.FORM_BASED_LOGIN_ENTRY_POINT.equals(path)) {
        response.setHeader("Access-Control-Allow-Headers",
                           "Origin, X-Requested-With, Content-Type, Accept,
                           authorization, username, password");
    } else {

```

```

        response.setHeader("Access-Control-Allow-Headers",
                            "Origin, X-Requested-With, Content-Type, Accept,
                            authorization");
    }

    if ("OPTIONS".equalsIgnoreCase(request.getMethod())) {
        response.setStatus(HttpServletResponse.SC_OK);
    } else {
        chain.doFilter(req, resp);
    }
}

```

En desarrollo esta clase está configurada para permitir cualquier header ya sea POST, GET, OPTIONS o DELETE y con diferentes headers permitidos. Sin realizar este filtro que añade los headers necesarios a la cabecera, ningún cliente web podría realizar peticiones al servidor debido a que cualquier navegador, por seguridad, las bloquearía.

6.6.4.2 Aplicación móvil

En este apartado se van a describir aquellas clases que se consideran más relevantes de la implementación de la aplicación móvil.

6.6.4.2.1 VolleyManager

Clase que implementa el patrón Singleton. Actúa como manager de las peticiones que se realizan al servidor. Contiene por tanto una queue con las peticiones que se van a enviar.

Retorno del método	Método y descripción
VolleyManager	getInstance (Context context) Retorna una instancia de la clase VolleyManager
RequestQueue	getRequestQueue () Retorna la queue con las peticiones pendientes de realización
void	addToRequestQueue (Request<T> req) Método genérico que añade una petición a la queue de peticiones

6.6.4.2.2 GsonRequest

Clase genérica para realizar peticiones HTTP y parsear la respuesta en JSON a un objeto del modelo de dominio. Obtenida parcialmente de la documentación oficial de Google y adaptada al dominio de la aplicación.

Retorno del método	Método y descripción
Map<String, String>	getHeaders () Método que envía en todas las peticiones la cabecera con el token de autenticación al servidor
byte[]	getBody() Retorna el body de la petición
getBodyContentType()	Retorna la constante “application/json; charset=utf-8”
void	deliverResponse (T response) Ejecuta el callback de respuesta del listener asociado a la petición.
Response<T>	parseNetworkResponse (NetworkResponse response) Método genérico que obtiene el objeto del modelo instanciado con la información retornada por el servidor.

6.6.4.2.3 ReportSender

Clase que extiende de BroadcastReceiver y que es ejecutada de manera periódica en el intervalo fijado por el usuario. Desde esta clase se inicia el proceso de obtener la posición del usuario y enviarla al servidor junto con todas las acciones que tenga pendientes.

Retorno del Método y descripción método	
void	<code>onReceive(Context context, Intent intent)</code> Método ejecutado por el BroadcastReceiver. Inicia la solicitud de actualización de la localización.
ReportState	<code>getConnection(Context context)</code> Retorna el objeto correspondiente al estado del dispositivo
void	<code>onLocationUpdate(Location location)</code> Método ejecutado desde la clase de obtención de localización cuando se actualiza. Obtiene el objeto asociado al estado del dispositivo y gestiona la localización en función de éste.
void	<code>onErrorLocationUpdate()</code> Ejecutado desde la obtención de la localización si ocurre algún error. Finaliza el broadcast.

6.6.4.2.4 LocationUtils

Clase utilizada para la obtención de actualizaciones de la posición GPS del dispositivo. Implementa LocationListener, GoogleApiClient.ConnectionCallbacks y ResultCallback<LocationSettingsResult> para poder solicitar las actualizaciones.

Retorno del Método y descripción método	
void	<code>onLocationChanged(Location location)</code> Método llamado cuando se obtiene la actualización de la posición del dispositivo.
boolean	<code>isGooglePlayServicesAvailable()</code> Retorna si es posible conectarse a los servicios de GooglePlay para obtener la posición
void	<code>startLocationUpdates()</code> Comienza el proceso de solicitud de localización llamando a la comprobación de settings del dispositivo.
void	<code>requestLocation()</code> Llama a LocationServices para obtener la localización del dispositivo
void	<code>onResult(LocationSettingsResult locationSettingsResult)</code> Método que maneja la respuesta de los settings del dispositivo.
void	<code>checkLocationSettings()</code> Realiza la petición para comprobar los settings del dispositivo acorde a la petición de localización.

Capítulo 7. Desarrollo de las Pruebas

7.1 Pruebas Unitarias

Durante el desarrollo del servidor se siguió la técnica *Test-driven development* (TDD) o Desarrollo guiado por pruebas. Esta técnica consiste en, básicamente, escribir primero las pruebas y después del código de la aplicación.

Al escribir primero las pruebas sin tener aún el código no pasan y una vez escrito el código se verifica que efectivamente funciona todo de manera correcta.

Gracias a usar TDD, entre otros beneficios, se consigue una mejor claridad de código ya que estás obligado a hacer unidades funcionales separadas de otras, con responsabilidades muy bien definidas. Además, si se sigue este método de forma correcta, se garantiza una alta cobertura de código.

No obstante, cabe destacar que, si las pruebas que se diseñan son de mala calidad o no cubren todas las situaciones que deberían, el uso de esta técnica puede ser inútil.

Mediante JUnit se han probado todos los servicios de negocio del servidor, los cuales son consumidos por los demás clientes. Cabe destacar, además, que el reducido número de pruebas unitarias es debido a que, en algunas ocasiones, sobre todo en ReportService, varios servicios son probados en el mismo método de prueba. Esto hace que el número total de test sea menor al que realmente podría ser.

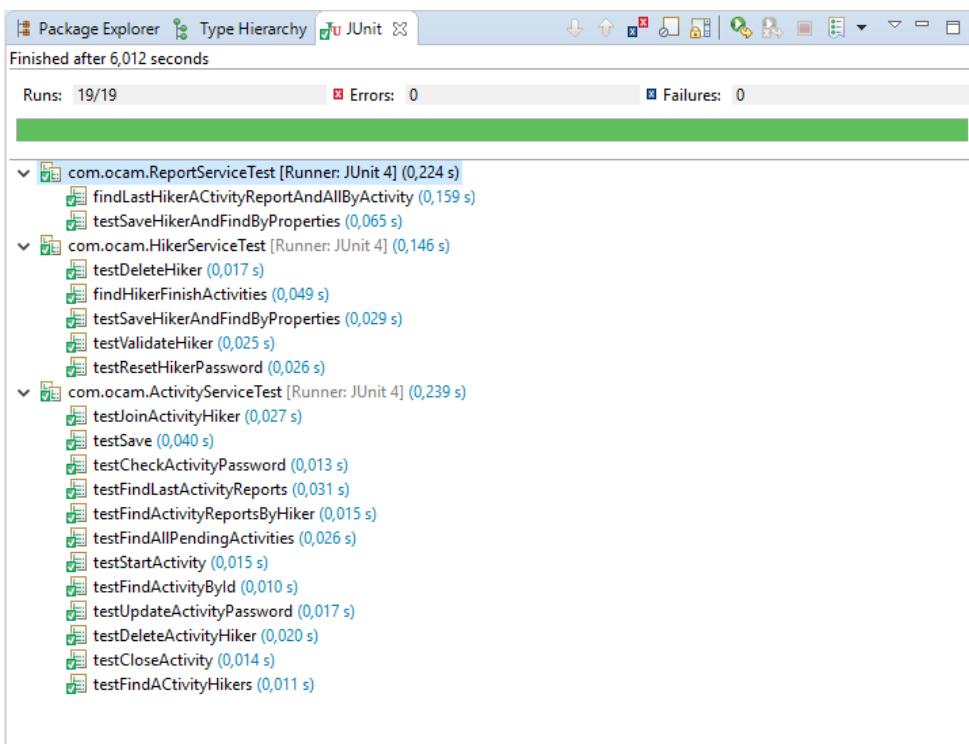


Figura 121 Pruebas JUnit servidor

7.2 Pruebas de comportamiento

Para realizar las pruebas de código de la aplicación web se ha utilizado la técnica de *Behavior-driven development* (BDD) o Desarrollo guiado por comportamiento. Esta técnica consiste en, básicamente escribir las pruebas como historias de usuario escribiendo en código los criterios de aceptación que quieras que se satisfagan.

Para la realización de estas pruebas se ha utilizado un framework BDD para Javascript, Jasmine, descrito anteriormente. Proporciona una manera sencilla de escribir las pruebas sin ser necesario manipular el árbol DOM.

Figura 122 Pruebas Jasmine aplicación web

Con estas pruebas se ha verificado el comportamiento de todos los métodos de los servicios que consumen datos del servidor, falseando las respuestas de éste mediante objetos mock. Los objetos mock son objetos simulados que imitan el comportamiento real.

El principal motivo de la utilización de este tipo de objetos para alguna de las pruebas de comportamiento es el de no supeditar la ejecución de éstas a que el propio servidor esté activo, o al estado de la red.

Además, se han probado otros escenarios con las actividades, listas de actividades, registro o login.

7.3 Pruebas de Integración y del Sistema

En este apartado se van a documentar los resultados de ejecutar todas las pruebas que se diseñaron en la fase de análisis, previa a la de implementación.

7.3.1 Subsistema página web

7.3.1.1 CU-W-1 Registrarse

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 1	En el sistema se habrá registrado con éxito la cuenta de usuario correo@gmail.com con password 2222\$2222	Sí	La cuenta aparece registrada pero aún no activada.
Caso de prueba 2	El sistema no debería registrar la nueva cuenta de usuario y debería notificar de que las contraseñas no coinciden.	Sí	El sistema no permite finalizar el proceso de registro.
Caso de prueba 3	El sistema no debería registrar la nueva cuenta de usuario y debería notificar de que el correo electrónico utilizado ya está registrado en el sistema.	Sí	El sistema notifica que el email ya está en uso por otro usuario.
Caso de prueba 4	El sistema no debería proceder con el registro de la cuenta y avisaría al usuario con el campo requerido que está en blanco.	Sí	El sistema no permite completar el registro y notifica del campo que falta por completar.

7.3.1.2 CU-W-2 Iniciar sesión

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 5	El sistema no debería permitir el inicio de sesión.	Sí	El sistema notifica de credenciales inválidas

7.3.1.3 CU-W-3 Ver actividades

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 6	El sistema mostrará las actividades registradas en el sistema, ordenadas decrecientemente por la fecha prevista de realización de la actividad.	Sí	El sistema muestra todas las actividades pendientes y en curso hasta el momento
Caso de prueba 7	El sistema no debería mostrar ninguna actividad	Sí	El sistema no muestra acceso para llegar a las actividades realizadas
Caso de prueba 8	La ordenación, tanto descendiente como ascendiente debería realizarse con normalidad y el idioma debería seguir siendo el inglés.	Sí	El sistema ordena las actividades según la columna seleccionada.

7.3.1.4 CU-W-4 Recuperar cuenta

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 9	Debería denegar el acceso a la aplicación puesto que la cuenta aún no ha sido validada.	Sí	El sistema notifica credenciales inválidas al intentar acceder.
Caso de prueba 10	El sistema debería bloquear el inicio de sesión advirtiendo que las credenciales son inválidas.	Sí	El sistema notifica credenciales inválidas al intentar acceder.

7.3.1.5 CU-W-5 Cerrar sesión

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 11	El sistema debería redireccionar a la pantalla de login y no reconocer el usuario logueado.	Sí	El sistema redirecciona a la pantalla de login y no permite acceder a las vistas privadas.

7.3.1.6 CU-W-6 Borrar cuenta

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 12	El sistema debería advertir de que las credenciales son inválidas.	No	El sistema permite iniciar sesión.
Caso de prueba 13	El sistema debería bloquear el registro y advertir de que el nombre de usuario está en uso.	Sí	El sistema informa de que el nombre de usuario ya está en uso.

7.3.1.7 CU-W-7 Crear actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 14	El sistema no debería registrar la nueva actividad y avisaría al usuario del campo que debe ser cumplimentado.	Sí	El sistema no permite completar el registro de la actividad.
Caso de prueba 15	El sistema no debería registrar la nueva actividad y avisaría al usuario de que el fichero de track de ruta .gpx no contiene datos.	Sí	El sistema informa de que el track es inválido.
Caso de prueba 16	El sistema no debería registrar la nueva actividad y avisaría al usuario de que el fichero de track de ruta .gpx no se puede procesar.	Sí	El sistema informa de que el track es inválido.
Caso de prueba 17	El sistema no debería permitir subir como fichero de ruta un archivo ejecutable.	Sí	El sistema informa de que el track es inválido.
Caso de prueba 18	El sistema advertiría de que no se ha introducido ningún guía para la actividad y no la guardaría.	Sí	El sistema no permite completar el registro
Caso de prueba 19	El sistema debería registrar la nueva actividad y en la página	Sí	El sistema redirecciona a la página de la actividad y en

	principal debería figurar para la fecha de realización introducida.		el listado sale la nueva actividad que acaba de ser creada.
--	---	--	---

7.3.1.8 CU-W-8 Editar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 20	El sistema no debería permitir acceder a la vista de edición de la actividad y avisando de que ya ha sido realizada.	Sí	El sistema no muestra clickable el acceso a la edición de la actividad
Caso de prueba 21	El sistema no debería actualizar la actividad y avisaría con los campos que son inválidos.	Sí	El sistema no muestra clickable el acceso a guardar los cambios sobre la actividad.
Caso de prueba 22	El sistema debería advertir al usuario que en una actividad en curso no se pueden eliminar guías ya existentes. No se habría guardado la edición de la actividad.	No	El sistema permite la eliminación del guía
Caso de prueba 23	El sistema no debería permitir la edición de la fecha de realización a una fecha ya pasada.	Sí	El calendario mostrado por el sistema no permite la selección de días pasados

7.3.1.9 CU-W-9 Cambiar datos personales

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 24	El sistema no debería permitir finalizar la edición puesto que faltan datos por completar. Al intentar acceder al sistema con la nueva contraseña el sistema debería notificar con credenciales inválidas al usuario y no permitir el acceso.	Sí	El sistema no permite completar los cambios al no haber introducido datos en todos los campos obligatorios.
Caso de prueba 25	El sistema debería entrar en sesión con la nueva contraseña.	Sí	Se puede volver a acceder con la contraseña nueva.

7.3.1.10 CU-W-10 Monitorizar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 26	El sistema no debería mostrar en el mapa nada más que el track esperado para la ruta.	Sí	No figura ningún excursionista en la tabla de participantes.
Caso de prueba 27	El sistema debería ocultar ambas polylineas del mapa y la posición de ambos excursionistas y posteriormente recargar los datos de la tabla, actualizando los existentes, y automáticamente volver a mostrar	Sí	Se recarga la información y las trayectorias que en ese momento son visibles sobre el mapa.

	ambas polilineas pero con los datos actualizados.		
Caso de prueba 28	La información del excursionista debería permanecer sobre el mapa en la misma posición y el mapa debería cambiar a Raster.	Sí	La trayectoria no varía, el mapa sí.
Caso de prueba 29	El sistema no debería mostrar en el mapa nada y el check y el botón del excursionista deberían estar desmarcados	Sí	No figura nada sobre el mapa y no hay cambios ni en el check de la posición ni en el botón de la trayectoria del excursionista.

7.3.1.11 CU-W-11 Eliminar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 30	El sistema debería mostrar inaccesible el acceso a eliminar la actividad.	Sí	El sistema no muestra clickable el acceso a eliminar la actividad
Caso de prueba 31	El sistema no debería listar la actividad borrada y por tanto sería inaccesible.	Sí	La actividad no aparece en ningún listado.

7.3.2 Subsistema aplicación móvil

7.3.2.1 CU-M-1 Registrarse

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 32	La aplicación debería avisar de que sin conexión a internet no se puede completar el registro.	Sí	No permite completar el registro
Caso de prueba 33	La aplicación no debería permitir el registro del usuario avisando que faltan datos por llenar.	Sí	Muestra un aviso al usuario informando de que es requerido llenar todos los campos
Caso de prueba 34	La aplicación deberá mandar la cuenta al servidor y vincularla con el dispositivo. Automáticamente deberá además ser redireccionado a la página principal	Sí	La aplicación notifica al usuario de que se ha enviado un correo electrónico de confirmación. La cuenta se ha registrado y es accesible

7.3.2.2 CU-M-2 Vincular cuenta

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 35	La aplicación deberá notificar al usuario que no hay conexión a internet y por tanto que no se puede realizar la vinculación.	Sí	La aplicación no permite la vinculación de la cuenta

Caso de prueba 36	La aplicación deberá no vincular la cuenta de usuario.	Sí	La aplicación informa de credenciales inválidas
Caso de prueba 37	Vincular la cuenta de usuario con el dispositivo y mostrar la lista de actividades pendientes. Además, en sucesivas ejecuciones de la aplicación, ésta no deberá volver a preguntar por las credenciales de usuario.	Sí	La aplicación inicia en sesión al usuario. Al reiniciarla directamente abre la ventana de actividades pendientes.
Caso de prueba 38	La aplicación debería iniciar en sesión al usuario advirtiéndole de que no tiene conexión y que los datos que se muestren pueden no ser los actualizados.	Sí	La aplicación informa de que no hay conexión a internet y que los datos podrían no ser los más actualizados. Muestra los datos del último inicio en sesión con conexión a internet.

7.3.2.3 CU-M-3 Ver actividades

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 39	La aplicación deberá mostrar en orden de fecha de realización las actividades disponibles en el sistema	Sí	Muestra todas las actividades ordenadas por fecha pendientes o en curso en el sistema
Caso de prueba 40	La aplicación deberá avisar al usuario de que se ha perdido la conexión con internet y deberá almacenar de forma local las actividades descargadas hasta el momento. Mostrará las actividades de la última vez que se recargaron los datos y no permitirá recargarlos más hasta no tener conexión a internet de nuevo en el dispositivo.	Sí	La aplicación no permite recargar las actividades e informa de que las que ve son las últimas guardadas con el último acceso a la aplicación con conexión a internet.
Caso de prueba 41	La aplicación deberá avisar de que no hay conexión a internet y que por tanto se muestran las actividades descargadas en el último acceso con conexión.	Sí	La aplicación informa de que las que ve son las últimas guardadas con el último acceso a la aplicación con conexión a internet y que actualmente no tiene conexión a internet en el dispositivo.

7.3.2.4 CU-M-4 Comenzar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 42	La aplicación deberá guardar en local el nuevo estado de la actividad y la contraseña introducida para	Sí	Al recargar la actividad con conexión a internet, aparece de la misma

	iniciarla. Al recuperar la conexión y recargar los datos de la actividad, ésta debería estar en estado en curso.		manera que cuando no había conexión, abierta.
Caso de prueba 43	La aplicación deberá enviar al servidor el estado de la actividad y la contraseña almacenados en local y figurar en la lista de actividades como actividad en curso.	Sí	La aplicación sigue de la misma manera que sin la conexión a internet
Caso de prueba 44	La aplicación no deberá dar por comenzada la actividad y avisará al guía de la misma de que la contraseña introducida para la actividad es inválida.	Sí	La aplicación no muestra clickable el campo de comenzar actividad
Caso de prueba 45	La aplicación deberá dar por comenzada la actividad y figurar accesible en la lista de actividades.	Sí	La aplicación redirecciona a la vista de monitorización y cambia el estado de la actividad a en curso.

7.3.2.5 CU-M-5 Monitorizar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 46	La aplicación deberá mostrar los datos de todos los excursionistas siendo monitorizados. Además, deberá guardar de forma local los datos de posición de cada excursionista cada vez que éstos se recarguen.	Sí	Muestra en la lista de excursionistas una entrada por cada excursionista unido en ese momento a la actividad con la hora del último reporte.
Caso de prueba 47	La aplicación deberá mostrar los datos de todas las personas que estaban siendo monitorizadas e incluir al nuevo excursionista.	Sí	Aparece una nueva entrada en la tabla con el nuevo excursionista unido a la actividad.
Caso de prueba 48	La aplicación deberá avisar al usuario de que la conexión a internet se ha perdido. Una vez recargada la actividad deberá recuperar y mostrar los datos almacenados de manera local.	Sí	No permite recargar los datos e informa de que no hay conexión a internet en el dispositivo.
Caso de prueba 49	La aplicación deberá descargar el nuevo estado de la actividad del servidor y notificar al usuario de que ha sido cerrada.	Sí	La actividad ya no sale en el listado de actividades pendientes por lo que no se puede acceder a la monitorización de la misma.

7.3.2.6 CU-M-6 Unirse a actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 50	La aplicación no deberá permitir unirse al usuario y mostrará al usuario una	Sí	La aplicación informa de un error uniéndose

	advertencia avisando de que la actividad no está en curso y por lo tanto no puede unirse a ella.		a la actividad, por lo que no se completa la acción
Caso de prueba 51	La aplicación no deberá permitir unirse al usuario y mostrará al usuario una advertencia avisando de que la actividad no está en curso y por lo tanto no puede unirse a ella.	Sí	La aplicación informa de contraseña errónea para la actividad
Caso de prueba 52	La aplicación deberá avisar al usuario de la pérdida de conexión a internet. La aplicación deberá obtener la posición GPS del dispositivo y almacenarla de manera local.	Sí	La aplicación informa en sucesivos reportes que han sido almacenados de forma local por falta de conexión a internet.
Caso de prueba 53	La aplicación deberá avisar al usuario de la reconexión a internet. Enviará al servidor y eliminará posteriormente todos los datos de posición almacenados de manera local. Obtendrá la posición GPS del dispositivo y la enviará al servidor.	Sí	La aplicación informa de que tiene reportes de posición guardados de manera local. Al reconnectarse pasan a enviados y se registran en el servidor.
Caso de prueba 54	La aplicación obtendrá la posición GPS del dispositivo y la enviará al servidor.	Sí	La aplicación informa de que se ha enviado el reporte y el servidor lo registra.
Caso de prueba 55	La aplicación deberá descargar el nuevo estado de la actividad y descartar, en caso de haberlos, reportes almacenados de manera local. Notificará mediante una advertencia en la pantalla al usuario de que la actividad ha sido cerrada y la monitorización ha concluido.	Sí	La aplicación envía al servidor todos los reportes encolados y no se guardan aquellos que tenían la fecha superior a la de cierre.
Caso de prueba 56	El sistema debería desechar los reportes de posición enviados tras el cierre de la actividad y debería notificar al usuario de que la actividad ha sido concluida.	Sí	En la monitorización de la actividad solo se encuentran los reportes previos a haberla cerrado.

7.3.2.7 Finalizar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 57	La aplicación deberá guardar de manera local el estado nuevo de la actividad y notificar al usuario de que se ha perdido la conexión a internet y por tanto no ha podido ser enviado al servidor.	Sí	La aplicación informa de que la actividad se ha cerrado de manera local y se enviará el nuevo estado al recuperar la conexión. Se detiene la monitorización del dispositivo.
Caso de prueba 58	La aplicación deberá enviar al servidor el nuevo estado	Sí	La actividad se cierra y se detiene la monitorización

	de la actividad. Notificará al usuario de que la actividad ha sido finalizada.		
--	--	--	--

7.3.2.8 Cambiar password

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 59	La aplicación deberá notificar tras el tiempo establecido de timeout que el cambio de contraseña no se ha podido llevar a cabo.	Sí	La aplicación no permite cambiar la contraseña y en BD figura la antigua.

7.3.2.9 Abandonar actividad

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 60	El sistema debería realizar las acciones que se llevaron a cabo sin conexión en el mismo orden. Deberían recogerse y registrarse todos los reportes enviados por el usuario y posteriormente dejar de ser monitorizado al abandonar la actividad	Sí	El servidor registra los tres reportes almacenados de manera local. Posteriormente el usuario abandona la actividad y no sigue siendo monitorizado.

7.3.2.10 Cerrar sesión

Caso de prueba	Resultado esperado	Pasa	Resultado obtenido
Caso de prueba 61	No debería iniciar sesión al usuario y por tanto se quedaría en la pantalla de login.	Sí	En sucesivas aperturas de la aplicación no inicia sesión y solicita los datos de nuevo al usuario.

7.4 Pruebas de Usabilidad

7.4.1 Subsistema aplicación web

Para realizar las pruebas de usabilidad de la aplicación web se ha realizado un formulario a un conjunto de usuarios no familiarizados con la aplicación.

En dicho formulario se pedía seguir una serie de pasos utilizando casi la totalidad de la funcionalidad de la aplicación, sin dar explicaciones de cómo llevar a cabo dichos pasos.

Todas las preguntas del formulario debían responderse de manera obligatoria aportando un valor en una escala de 1 a 10 ambos inclusive.

- 1) Accede al listado de actividades y recarga los datos de las actividades. ¿Con qué dificultad te has encontrado para hacerlo?
- 2) Ordena las actividades del listado según la fecha prevista descendiente ¿Con qué dificultad te has encontrado para hacerlo?
- 3) Intenta acceder al detalle de una actividad y vuelve al login ¿Con qué dificultad te has encontrado para hacerlo?
- 4) Inicia sesión con el usuario "formulario" y password "formulario". Posteriormente desde la lista de actividades cambia el idioma a inglés, observa que la página se ha traducido, y luego cámbialo de nuevo a español. ¿Con qué dificultad te has encontrado para hacerlo?
- 5) Crea una nueva actividad. Para hacerlo introduce valores en todos los campos obligatorios. Descarga el track de la ruta de aquí: <https://www.dropbox.com/s/reks37yp1szrdpo/track.gpx?dl=0> (lo necesitarás para crear la actividad). ¿Con qué dificultad te has encontrado para hacerlo?
- 6) Vuelve a la lista de actividades y accede al detalle de la actividad que acabas de crear. ¿Con qué dificultad te has encontrado para hacerlo?
- 7) Edita la actividad que has creado. Añade el guía de nombre "victor". Guarda. ¿Con qué dificultad te has encontrado para hacerlo?
- 8) En el detalle de la actividad que has creado cambia el mapa a tipo OSM. ¿Con qué dificultad te has encontrado para hacerlo?
- 9) Elimina la actividad que has creado. ¿Con qué dificultad te has encontrado para hacerlo?

- 10) Vuelve a la lista de actividades. Accede a la actividad de descripción "Actividad formulario usabilidad". Accede a la monitorización de la actividad, la password es 1234. ¿Con qué dificultad te has encontrado para hacerlo?
- 11) Desde la monitorización. Cambia el mapa a "OSM". Selecciona el usuario "víctor" y observa su posición en el mapa. ¿Con qué dificultad te has encontrado para hacerlo?
- 12) Desde la monitorización. Muestra y oculta la trayectoria del usuario "victor". Haz y deshaz zoom sobre el mapa para ver los puntos exactos por los que ha pasado. ¿Con qué dificultad te has encontrado para hacerlo?
- 13) Cierra sesión. ¿Con qué dificultad te has encontrado para hacerlo?

La media de notas obtenidas al realizar el test a un total de 23 personas, para cada una de las preguntas es:

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Media	9,31	9,35	9,43	9,59	8,95	9,86	9,50	9,48	9,49	9,53	9,58	9,49	9,66

Tabla 17 Notas medias preguntas formulario usabilidad



Figura 123 Gráfico de barras media de notas preguntas formulario usabilidad

De la obtención de estas notas, y consultas verbales a los encuestados, se recogieron las siguientes sugerencias, que además se llevaron a cabo:

Sugerencia	Acciones realizadas
Puede llegar a ser difícil de localizar, o no trivial, los botones de eliminar, monitorizar y editar una actividad al estar contenidos de un mismo botón en la cabecera de una actividad.	Se eliminó el botón y se sacaron los otros tres botones (monitorizar, editar y eliminar) a la ventana de la actividad de una manera visible y fácil de localizar.
Resaltar los estados de la actividad para poder distinguirlos de manera sencilla y así saber en cual entrar si se identifica la actividad como "la que está en curso".	Se aumentó el tamaño del texto del estado de la actividad en dos puntos y se resaltó mediante colores: verde en curso y rojo pendiente.
Aumentar la información a cerca del enlace MIDE. Cuando introduces un enlace que es	Se corrigió dicho mensaje añadiendo un ejemplo de lo que es un enlace MIDE.

inválido la información aportada por la web es “Enlace MIDE inválido.”. Dar más datos en este aspecto.	
Resaltar determinados errores. Cuando introduces las credenciales en el acceso a la aplicación o bien un nombre de guía inválido en el formulario para dar de alta una actividad, los errores son texto plano sin color ni tamaño. Resaltar esto para que se vea de primera al intentar confirmar los cambios/hacer login.	Se introdujo un recuadro de error para dichos errores incluyendo el texto de los mismos dentro de dicho recuadro.

Tabla 18 Sugerencias usabilidad subsistema web

7.4.2 Subsistema aplicación móvil

Para la aplicación móvil se siguió el mismo procedimiento que con la aplicación web. Se realizó un formulario el cual fue realizado por usuarios no familiarizados con la aplicación. Una vez finalizado el formulario se preguntó por sugerencias con el fin de hacer la aplicación más usable.

Todas las preguntas del formulario debían responderse de manera obligatoria aportando un valor en una escala de 1 a 10 ambos inclusive.

- 1) Inicia la aplicación. Loguea con el usuario "formulario" password "formulario" (al ser la primera conexión puede tardar hasta un minuto). Una vez iniciado sesión recarga la lista de actividades. ¿Con qué dificultad te has encontrado para hacer esto?
- 2) Filtra las actividades. Muestra solamente aquellas que comiencen a partir del 20/05/2017. Observa que solamente podría quedar una en la lista. Recarga la lista para que vuelvan todas. ¿Con qué dificultad te has encontrado para hacer esto?
- 3) Accede a la configuración. Cambia el tiempo de refresco a 10 minutos. Vuelve a la lista de actividades. ¿Con qué dificultad te has encontrado para hacer esto?
- 4) Cambia la contraseña de la cuenta. Escribe como nueva contraseña "formulario" (es también la actual). ¿Con qué dificultad te has encontrado para hacer esto?
- 5) Accede a una actividad y posteriormente al track de la ruta. Cambia el tipo de mapa a "MapBoxSatellite". ¿Con qué dificultad te has encontrado para hacer esto?
- 6) Accede a la monitorización de la actividad "Actividad formulario usabilidad" y pon como tipo de mapa "MapBoxTerrain". ¿Con qué dificultad te has encontrado para hacer esto?
- 7) Recarga los datos de los excursionistas siendo monitorizados. ¿Con qué dificultad te has encontrado para hacer esto?

- 8) Muestra sobre el mapa la posición de "alejandro" y quítala después. ¿Con qué dificultad te has encontrado para hacer esto?

- 9) Cierra sesión. ¿Con qué dificultad te has encontrado para hacer esto?

Al igual que con el formulario de usabilidad de la aplicación web, de la obtención de estas notas, y consultas verbales a los encuestados, se recogieron las siguientes sugerencias, que además se llevaron a cabo:

Sugerencia	Acciones realizadas
Hay determinados iconos, como el del track de la ruta y el de las capas del mapa que no son demasiado identificativos. Solamente con decir “accede al track de la ruta” y viendo ese ícono, puede ser difícil de intuir que ese ícono es del track, y lo mismo para los tipos de mapa.	Se sustituyeron ambos iconos por dos nuevos utilizados comúnmente por aplicaciones web y móviles relacionadas con mapas y capas sobre mapas.
Al monitorizar una aplicación, si sitúas dos excursionistas sobre el mapa no se sabe quién es quién a menos que hagas click sobre sus respectivos markers. Se deberían distinguir por colores.	Se asignó un color de manera automática y aleatoria a cada excursionista unido a una actividad. De esta forma, estableciendo el mismo color al nombre del hiker en la lista se pueden identificar sobre el mapa a simple vista.

Tabla 19 Sugerencias usabilidad subsistema aplicación móvil

7.5 Pruebas de Accesibilidad

Tanto la aplicación móvil como la aplicación web han sido diseñadas para garantizar que, de manera accesible, cualquier usuario pueda hacer uso de ellas de manera independiente a los dispositivos que se utilicen, o las circunstancias de los mismos.

Antes de comenzar con las pruebas, se ha decidido no justificar un nivel de conformidad doble A debido a que la página web, resumiendo, no se encuentra entre los siguientes casos:

- Página web pública servida por administración pública
- Empresa financiada con capital público
- Empresa privada con más de 100 empleados y facturación superior a 60.000€

Por ello en estas pruebas se va a comprobar las versiones de navegador en las que la aplicación web es funcional y la responsividad tanto de ella como de la aplicación móvil.

7.5.1 Subsistema aplicación web

Para la realización de las pruebas de accesibilidad de la aplicación web se va a ejecutar en **diferentes versiones de diferentes navegadores** y posteriormente se mostrará **la misma ventana en diferentes resoluciones** para probar la responsividad.

7.5.1.1 Pruebas de versiones en navegadores

Se comprobará la **versión mínima y máxima** en la que la aplicación funciona de manera correcta en cada uno de los navegadores sobre los que se realicen las pruebas.

A continuación, se muestran capturas de la aplicación web funcionando en los navegadores en versiones mínimas y máximas.

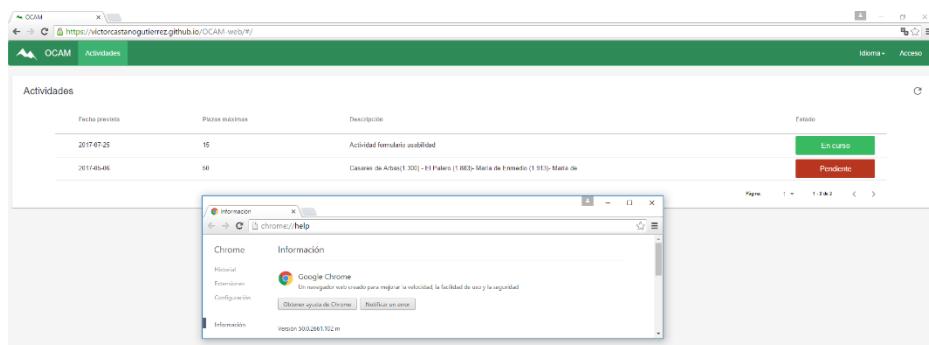


Figura 124 Aplicación web en chrome versión 50.0.266

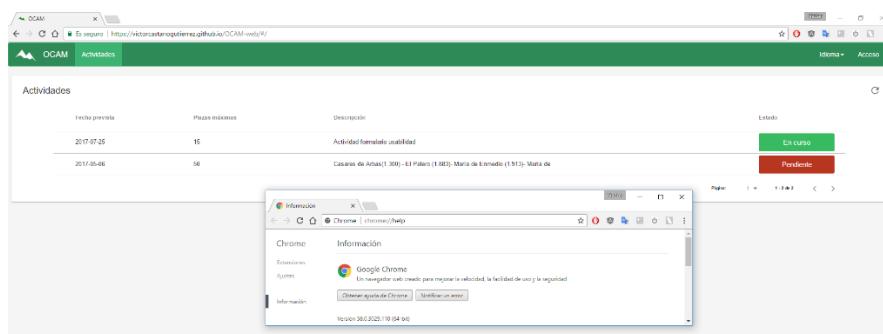


Figura 125 Chrome versión 58.0.3029

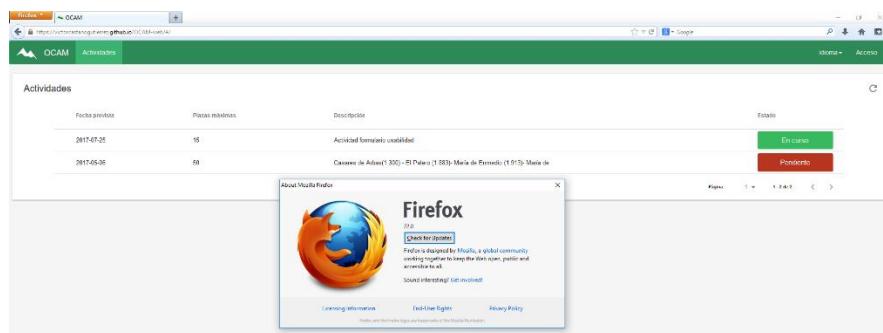


Figura 126 Aplicación web en Firefox 22.0

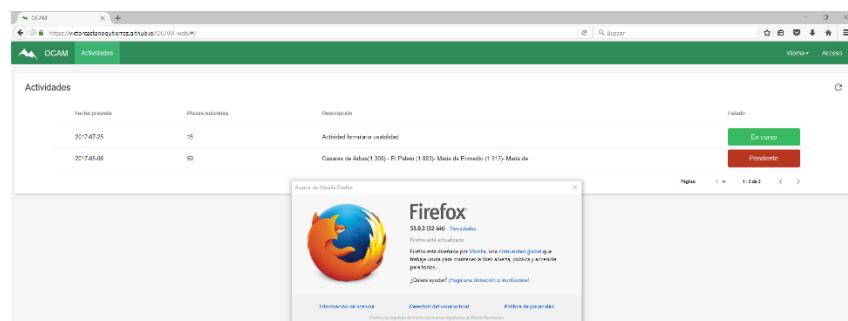


Figura 127 Aplicación web en Firefox 53.0.3

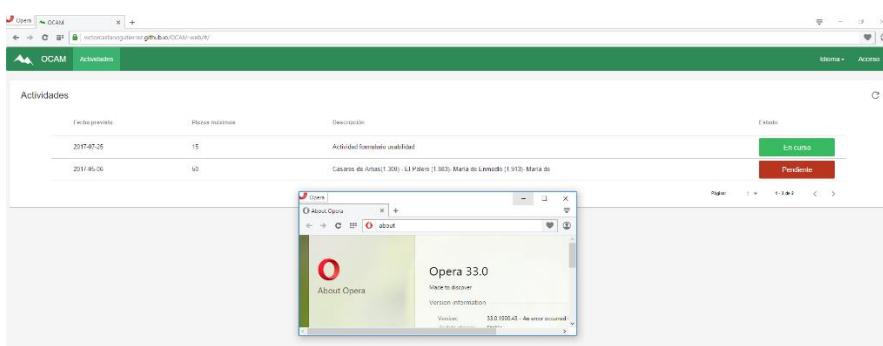


Figura 128 Aplicación web en Opera 33.0

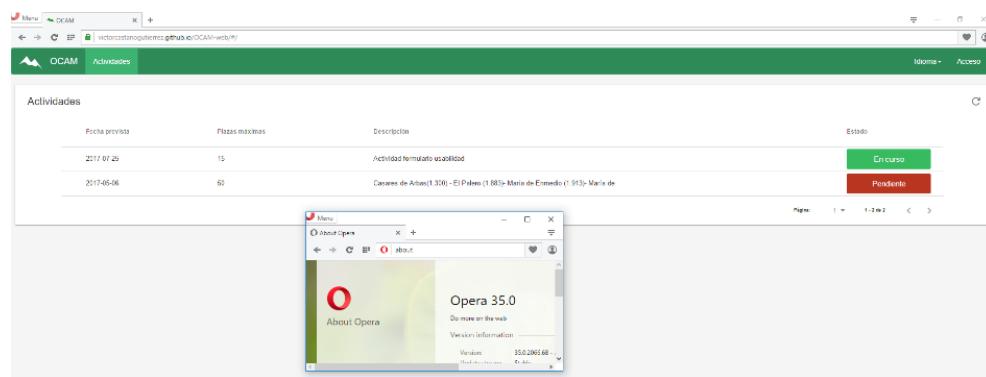


Figura 129 Aplicación web en Opera 35.0

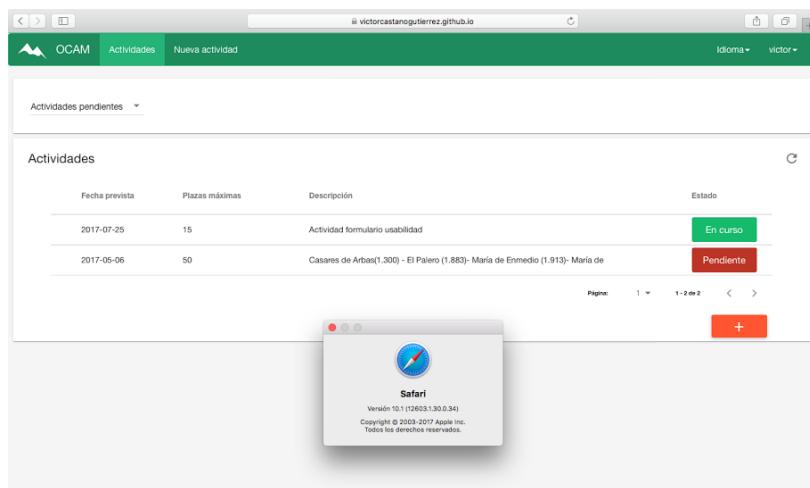


Figura 130 Aplicación web en Safari 10

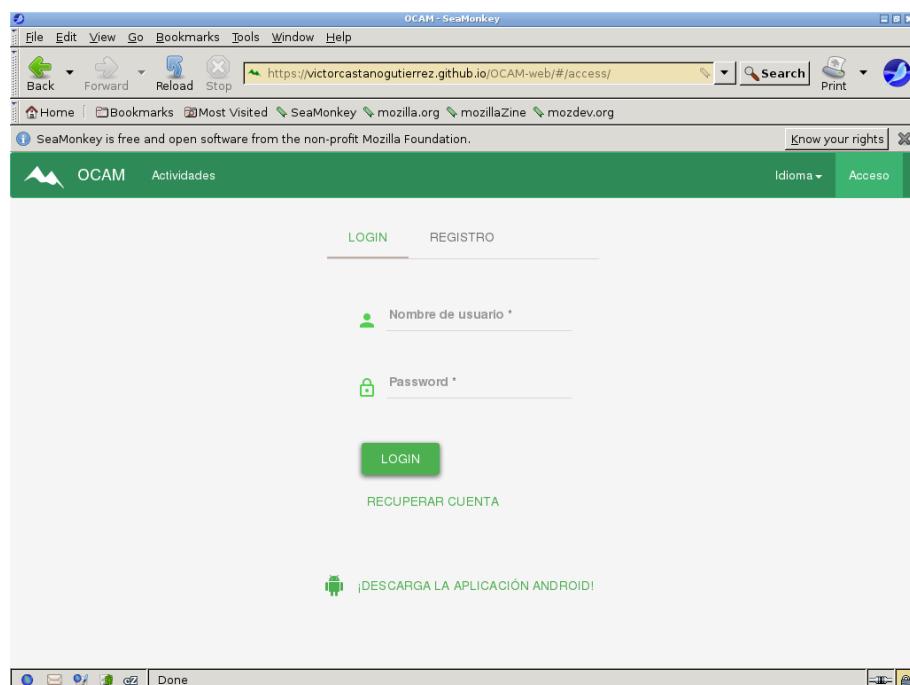


Figura 131 Aplicación web en SeaMonkey 2.22.1

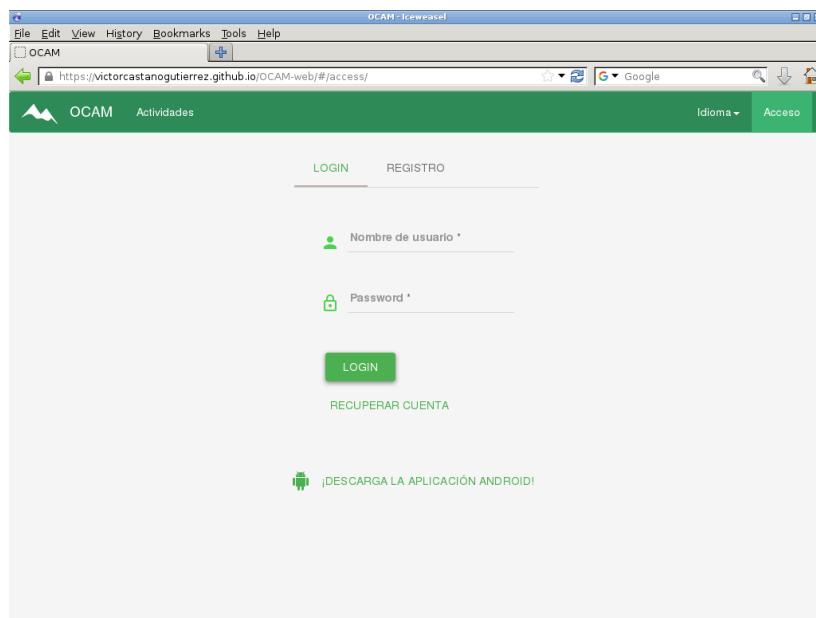


Figura 132 Aplicación web en SeaMonkey

Finalmente, en la siguiente tabla se muestran las versiones de los navegadores en los que se puede ejecutar la aplicación web.

Navegador	Versión mínima
Google Chrome	50
Mozilla Firefox	22
Opera	33
SeaMonkey	2.22.1
Iceweasel	24.3.0

Tabla 20 Versiones de navegadores – aplicación web

7.5.1.2 Pruebas de responsividad

Para la realización de las pruebas de responsividad se va a visualizar dos vistas de la aplicación web con diferentes componentes visuales en cada una de ellas, en dos diferentes resoluciones: resolución de PC y resolución móvil. Se comenzará observando la lista de actividades en una resolución de 1920x1080.

Fecha prevista	Plazas máximas	Descripción	Estado
2017-09-23	15	Actividad formulario usabilidad	En curso
2017-05-06	50	Casares de Arbas(1.300) - El Palero (1.883)- María de Enmedio (1.913)- María de	Pendiente

Figura 133 Lista de actividades con alta resolución

A continuación, haciendo uso de las herramientas que aporta el propio navegador se probará la misma vista en una resolución de dispositivo móvil de 412x732.

Fecha prevista	Plazas máximas	Descripción
2017-09-23	15	Actividad formulario usabilidad
2017-05-06	50	Casares de Arbas(1.300) - El Palero (1.883)- María de Enmedio (1.913)- María de

Figura 134 Listado de actividades con resolución 412x732

Vemos como la página se ha adaptado la resolución y no se ha perdido visibilidad, teniendo disponible en scroll horizontal todo el contenido de la tabla. Además, en el caso de la toolbar en la parte superior de la pantalla ha cambiado de formato pudiendo ocultarse como un menú desplegable. Vemos en la siguiente imagen, con el dispositivo rotado (732x412) lo que ocurre.

Fecha prevista	Plazas máximas	Descripción	Estado
2017-09-23	15	Actividad formulario usabilidad	En curso
2017-05-06	50	Casares de Arbas(1.300) - El Palero (1.883)- María de Enmedio (1.913)- María de	Pendiente

Figura 135 Listado de actividades con resolución 732x412

A continuación, se muestra otra de las vistas de la aplicación en las dos resoluciones mostradas anteriormente para el listado de actividades. En este caso, la monitorización.

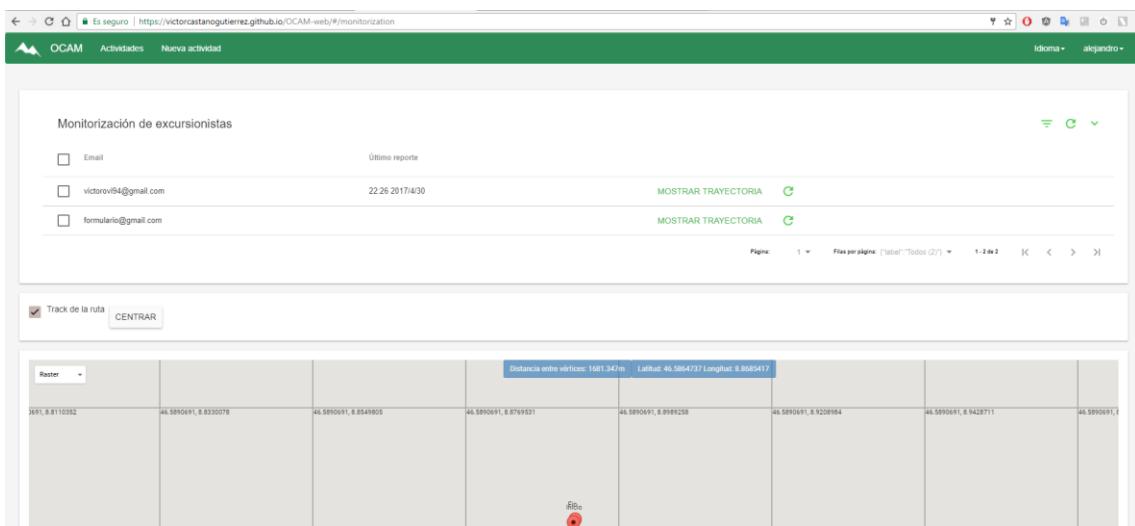


Figura 136 Vista de monitorización con resolución 1920x1080

La pantalla para mostrar el contenido se divide en tres *cards*. Cada uno de ellos muestra una parte del contenido de la vista. Ahora con resolución móvil, igual que la anterior vista:

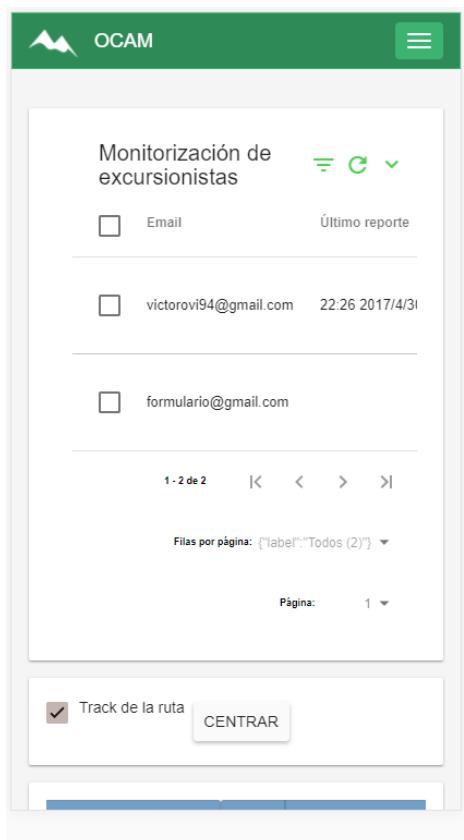


Figura 137 Monitorización en resolución 412x732

Tal y como se aprecia en la figura anterior, los *cards* de la pantalla son reordenados y situados uno detrás de otro de manera vertical. De esta manera, solamente haciendo scroll en el mismo eje se puede ver todo el contenido de la página. En este caso, y similar a la anterior vista, la toolbar sufre el mismo cambio.

7.5.2 Subsistema aplicación móvil

7.5.2.1 *Pruebas de responsividad*

Para probar la responsividad de la aplicación móvil se va a emular la vista del listado de actividades en diferentes resoluciones.

La primera prueba se realiza con un Nexus 6P de resolución 1440x2560dpi en orientación de pantalla vertical.



Figura 138 Listado de actividades móvil en 1440x2560dpi

A continuación, se va a realizar la misma prueba en un dispositivo con mayor resolución, Tablet Nexus 10 2560x1600dpi, también en orientación vertical.

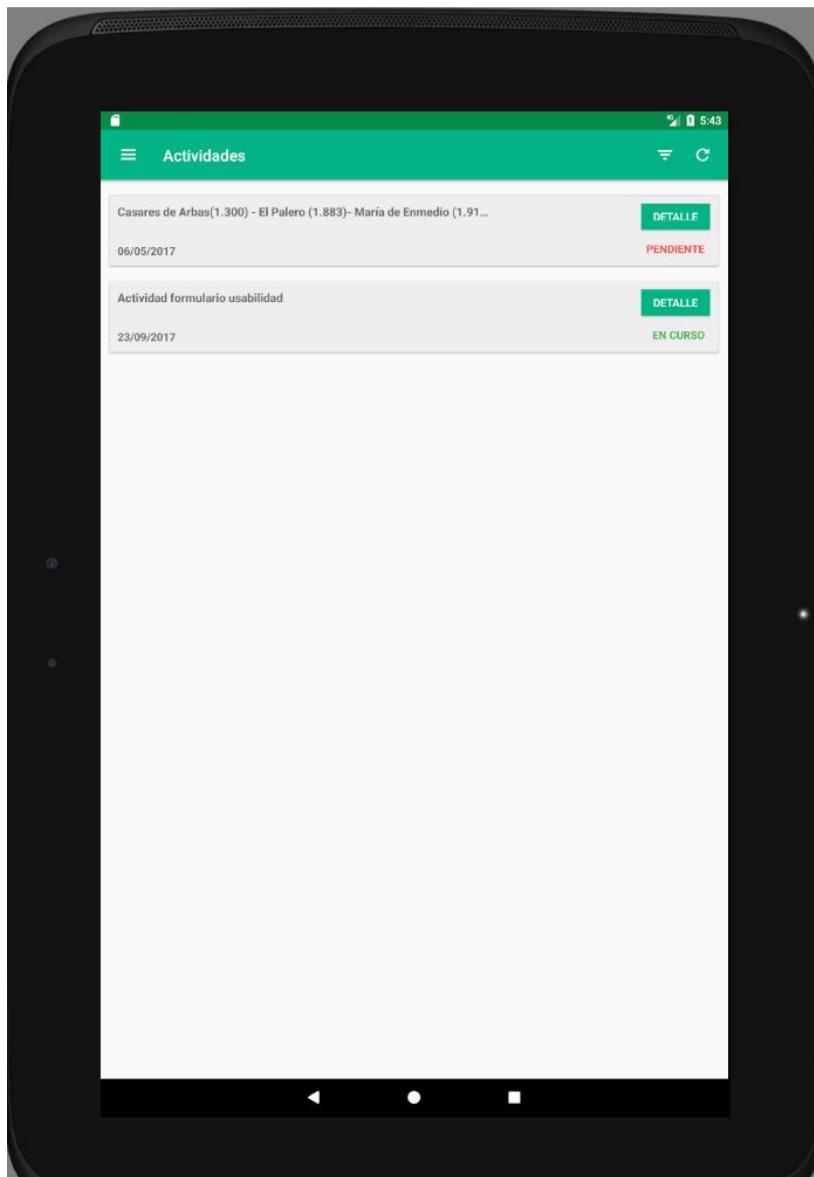


Figura 139 Listado de actividades móvil en 2560x1600dpi

Vemos como en ningún caso ningún componente de la vista queda oculto a raíz de cambios de resolución, ni para a menor ni para la mayor.

7.6 Pruebas de Rendimiento

Las pruebas de rendimiento se han realizado en las siguientes condiciones:

- Se ha enviado siempre la misma petición GET contra el servidor a la ruta “pendingActivities/0/1000” que es la que devuelve el listado de actividades, con sus correspondientes tracks.
- El servidor está alojado en Heroku, una plataforma de hosting con, entre otros, planes limitados gratuitos. El servidor cuenta con uno de estos planes gratuitos.
- Las pruebas se han realizado utilizando el software gratuito Apache JMeter.

7.6.1 Primera prueba

Se va a realizar un testo de rendimiento sobre el servidor, en una ruta en concreto para listar actividades. Se mandarán un total de 90 peticiones contenidas en un segundo con el fin de analizar si el servidor es capaz de atenderlas todas.

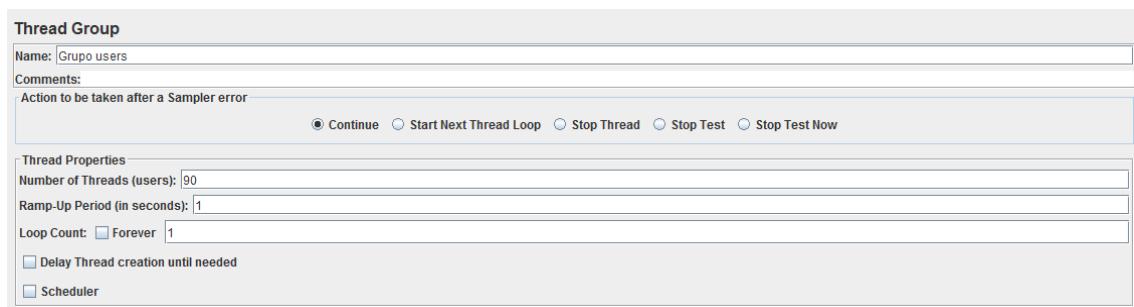


Figura 140 Grupo de usuarios primera prueba rendimiento – Jmeter

En la figura anterior podemos observar la configuración del grupo de usuarios que se mandarán al servidor. 90 usuarios en un segundo una sola vez.

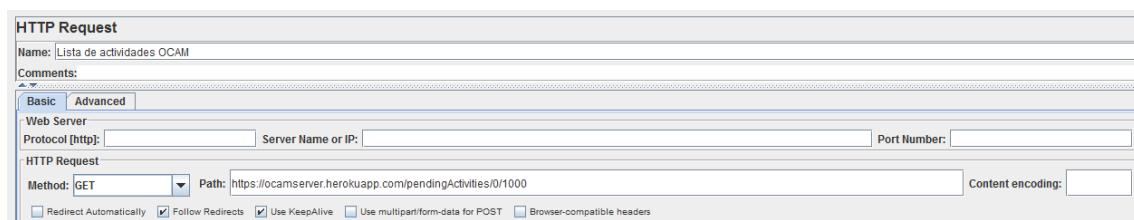


Figura 141 Petición HTTP realizada primera prueba rendimiento JMeter

Tal y como se observa en la anterior figura, se realiza la configuración para que cada usuario realice una petición GET a la URL del servidor.

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	<input type="text"/>				<input type="button" value="Browse..."/>	<input type="checkbox"/> Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Lista de actividad...	90	522	473	579	25.03	0.00%	60.2/sec	4879.52	8.87	83055.0
TOTAL	90	522	473	579	25.03	0.00%	60.2/sec	4879.52	8.87	83055.0

Figura 142 Resultados primera prueba rendimiento – Jmeter

Tal y como se observa en la anterior figura, el resumen de la prueba, hemos obtenido un 0,00% de error, esto quiere decir que ninguna petición no ha sido respondida con un código HTTP 200.

Por lo tanto, vamos a realizar una segunda prueba para intentar llegar al límite donde no podría aceptar más peticiones simultaneas el servidor con el fin de acercarlo a una situación de carga real.

7.6.2 Segunda prueba

Las condiciones a las que se ha sometido el servidor con la primera prueba son de 90 peticiones HTTP GET, a la ruta especificada, contenidas todas en un intervalo de 1 segundo.

El porcentaje de error, es decir, el porcentaje de peticiones sin respuesta 200 por parte del servidor en este caso es de 0%. Es decir, todas las peticiones son atendidas y resueltas.

Se va a realizar una segunda prueba aumentando el número de peticiones en 10, en el mismo intervalo de tiempo, es decir, 100. La configuración es muy similar a la de la primera prueba exceptuando el grupo de usuarios.

Thread Group										
Name:	Grupo users									
Comments:										
Action to be taken after a Sampler error	<input checked="" type="radio"/> Continue <input type="radio"/> Start Next Thread Loop <input type="radio"/> Stop Thread <input type="radio"/> Stop Test <input type="radio"/> Stop Test Now									
Number of Threads (users):	100									
Ramp-Up Period (in seconds):	1									
Loop Count:	<input type="checkbox"/> Forever <input checked="" type="checkbox"/> 1									
<input type="checkbox"/> Delay Thread creation until needed										
<input type="checkbox"/> Scheduler										

Figura 143 Grupo de usuarios segunda prueba rendimiento JMeter

Como se puede observar en la anterior figura, para esta prueba ha sido aumentado a 100, con la misma frecuencia.

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	<input type="text"/>				<input type="button" value="Browse..."/>	<input type="checkbox"/> Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Lista de actividad...	100	481	450	553	18.88	0.00%	68.4/sec	5547.77	10.09	83055.0
TOTAL	100	481	450	553	18.88	0.00%	68.4/sec	5547.77	10.09	83055.0

Figura 144 Resultados segunda prueba rendimiento JMeter

Como se puede apreciar en la figura de resultados, se obtiene un error del 0%, por lo que como se comentó anteriormente, el servidor es capaz de resolver todas las peticiones.

7.6.3 Tercera prueba

Vamos a aumentar, de nuevo, en 10 el número de peticiones que se realizarán para esta prueba. El intervalo de tiempo será el mismo que en pruebas anteriores, es decir, 1 segundo.

La configuración de JMeter es similar a la de pruebas anteriores por lo que se pasa a exponer la figura de resultados.

Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename <input type="text"/>										
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Lista de actividad...	110	676	420	1136	155,19	36,36%	63,7/sec	3312,17	9,40	53218,3
TOTAL	110	676	420	1136	155,19	36,36%	63,7/sec	3312,17	9,40	53218,3

Figura 145 Resultados tercera prueba rendimiento JMeter

Tal y como se observa en la figura, el servidor no ha sido capaz de responder a todas las peticiones. En total, un 36,36% de ellas han sido respondidas con el código 500, error de servidor. Esto es debido a que la memoria del servidor de aplicaciones se agota y no es capaz de seguir sirviendo peticiones.

Para ver de una manera más clara los resultados de esta prueba que han llevado al servidor a producir errores, se van a ilustrar mediante gráficas.

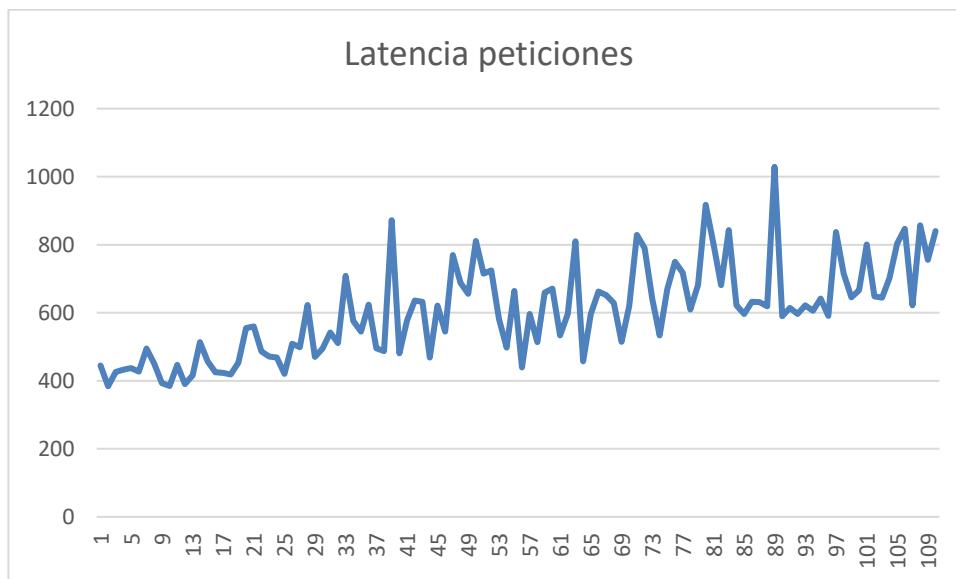


Figura 146 Latencia de las peticiones tercera prueba rendimiento JMeter

En la figura anterior se observa la latencia de todas las peticiones que se enviaron al servidor. Se puede ver una tendencia en aumento, desde las primeras con 400ms hasta alguna incluso sobrepasar 1000ms.

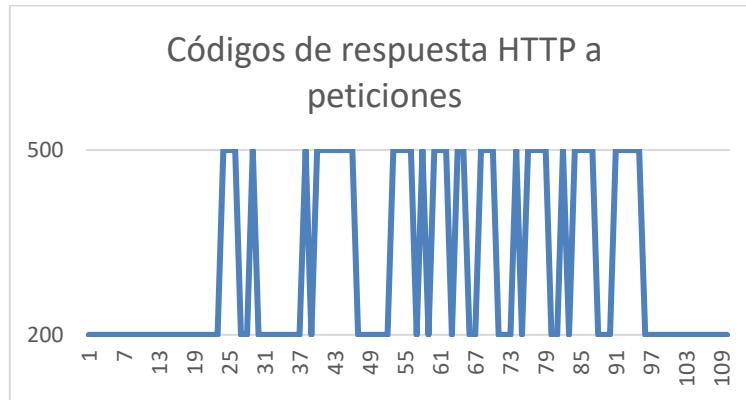


Figura 147 Códigos de respuesta HTTP peticiones prueba 3 rendimiento JMeter

Tal y como se puede observar en la anterior figura, ilustra de manera similar a los tiempos de respuesta lo que sucedió con la tercera prueba. A medida que se iban haciendo peticiones la tendencia a respuestas con código 500 iba aumentando.

7.6.4 Conclusiones

Número de peticiones	Intervalo (segundos)	% de error
90	1	0%
100	1	0%
110	1	36,36%

Tabla 21 Resumen pruebas rendimiento

Tal y como se aprecia en la figura anterior, y obtenida la información de las tres pruebas realizadas, vemos que aproximadamente, **el número de peticiones que en un segundo es capaz de atender el servidor es de 100**.

Esto nos sirve para poder acercar estos resultados a una situación real de carga, y saber, en función de dicha situación, si el servidor puede soportarla o no.

Vamos, por tanto, a extrapolar una situación real que se ajuste al máximo de peticiones que puede soportar el servidor:

- **100 actividades en curso** de manera simultánea con una media de 10 excursionistas en cada una: **1000 excursionistas**.
- Cada excursionista (algo improbable), envía **un reporte de posición cada minuto**.

Con estas características, tendríamos aproximadamente, 1000 peticiones por minuto. Además, hay que añadir las personas que están monitorizando. Supongamos una persona por actividad, **100 personas, refrescando la página** una vez cada minuto para conocer la posición de los participantes.

Por tanto, tendríamos, sumadas a las 1000 anteriores, **1100 peticiones por minuto**, en total, aproximadamente.

Esto es, teniendo un total de 1100 peticiones por minuto, y suponiendo que estén distribuidas de manera plana por segundo tendremos aproximadamente **18 peticiones por segundo**, lo que teniendo en cuenta los resultados de las pruebas anteriores es totalmente asumible por el servidor.

Siguiendo la misma distribución, podríamos asumir hasta 545 actividades simultaneas, basándonos en:

*10 excursionistas (1 reporte por minuto cada uno) * actividades + 1 monitorizador (1 actualización por minuto) * actividades <= 6000 peticiones / minuto máximo para el servidor*

Actividades máximas: 545 (siempre bajo una distribución normal)

Si la distribución de las peticiones en cada minuto no fuese normal, haciendo referencia al primer escenario planteado, habría aún un margen bastante amplio de 42 peticiones (60-18) en cada minuto que el servidor sí podría resolver.

Capítulo 8. Manuales del Sistema

8.1 Manual de Instalación

8.1.1 Instalación del servidor

Lo primero que se necesita para la instalación del servidor y la posterior ejecución es los ficheros del proyecto. Entre ellos, será de vital importancia el certificado HTTPS con el que arrancará la aplicación.

El fichero *keystore.p12* está suministrado junto a los archivos del servidor y deberá estar localizado en la raíz del mismo para poder instalarlo.

Para la instalación del servidor con la base de datos existen varios perfiles posibles a seguir. Cada perfil está situado en el directorio *src/main/resources*. Actualmente hay configurados tres diferentes perfiles: *prod*, *prodPostgre* y *test*. Se pueden configurar tantos como sean necesarios.

En el fichero *application.properties* del citado directorio se encuentra el perfil sobre el que arrancará la aplicación. En el caso de indicar *prodPostgre*, la aplicación utilizará el fichero de configuración *application-prodPostgre.properties* para arrancar.

- ProdPostgre: perfil que utiliza una base de datos PostgreSQL en Heroku.
- Prod: perfil que utiliza una base de datos MySQL que está en línea.
- Test: perfil que utiliza una base de datos interna. Esto es, se crea al iniciar la aplicación y se destruye al pararla.

Si se quisiera instalar el servidor para utilizar una base de datos propia, no haría más que alterar los ficheros de configuración proporcionados. Si el gestor de base de datos que se quiere usar no es ni MySQL ni PostgreSQL, sería recomendable crear un nuevo perfil con la configuración deseada.

El puerto por defecto es el 8443 y el perfil *prodPostgre*. Para cambiar de perfil, alteramos el campo *spring.profiles.active* al nombre del perfil (cadena después del guion en el nombre del fichero de propiedades).

```
spring.profiles.active: prodPostgre
spring.application.name: Outdoor Colective Activities Monitorization
server.port: 8443
```

Se recomienda no alterar el puerto de la aplicación, puesto que esto conllevaría a necesitar realizar cambios en la configuración de la aplicación móvil y web.

8.1.2 Aplicación móvil

El único requisito para instalar la aplicación móvil en cualquier dispositivo es el de, teniendo la versión adecuada del sistema operativo Android, permitir la instalación de aplicaciones desde orígenes desconocidos.

Típicamente esta configuración puede encontrarse en Ajustes > Orígenes desconocidos. Tendrá que estar activado si se quiere instalar el APK aportado con los ficheros del proyecto en el dispositivo.

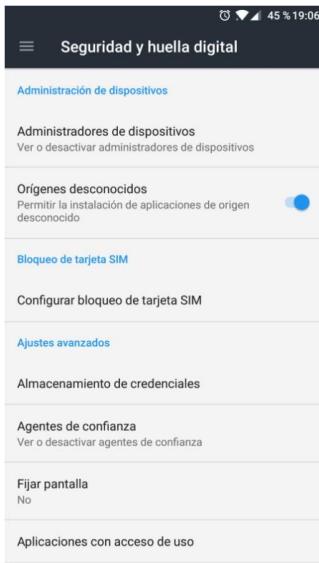


Figura 148 Instalación de aplicaciones de orígenes desconocidos

La instalación de la aplicación móvil, por defecto, está configurada para realizar peticiones contra el servidor desplegado en Heroku. Si quisieramos utilizar nuestro propio servidor sería necesario alterar la clase `com.ocam.util.Constants.java`. En esta clase se encuentra la constante `SERVER_URL` donde indicaremos la dirección del nuestro. Esto requerirá, además, recompilar la aplicación.

8.1.3 Aplicación web

Para la instalación de la aplicación web se requiere lo mismo que para la aplicación móvil. No será necesario alterar ningún parámetro a no ser que queramos que realice las peticiones contra nuestro propio servidor. Por defecto, lo hará contra el desplegado en Heroku.

Si alterásemos el servidor habría que introducir la URL en el fichero localizado en la ruta `src/common/common.module.js` donde se encuentran las constantes entre la cual está `SERVER_URL`.

8.2 Manual de Ejecución

Por defecto el proyecto se encuentra ejecutado en las siguientes URL:

- Servidor: <https://ocamserver.herokuapp.com>
- Página web: <https://victorcastanogutierrez.github.io/OCAM-web/> (puede tardar hasta dos minutos en cargar solo la primera vez).
- Aplicación móvil: se puede descargar desde <https://victorcastanogutierrez.github.io/OCAM-web/#/access> o bien instalarla desde el APK entregado con los ficheros del proyecto.

Si se quisiera ejecutar el servidor, tras realizar las configuraciones deseadas habría varias opciones:

- Ejecutarlo desde el entorno de desarrollo

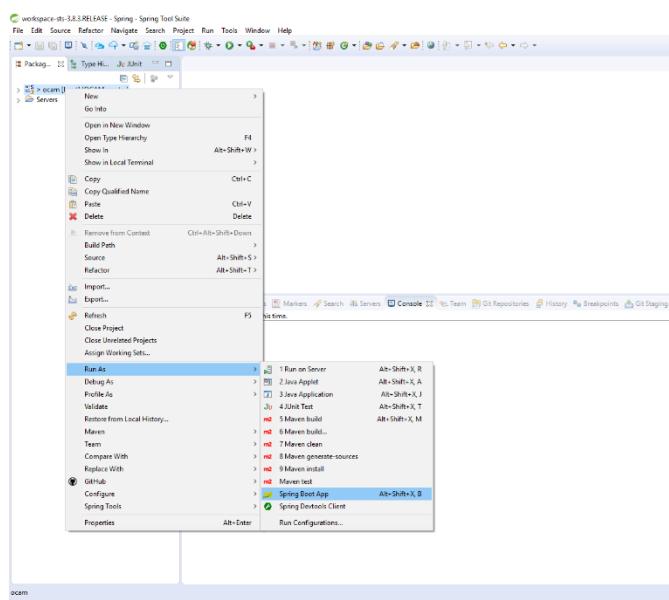


Figura 149 Ejecución del servidor desde entorno de desarrollo

- Exportar el proyecto como un war para su posterior ejecución en otro entorno:

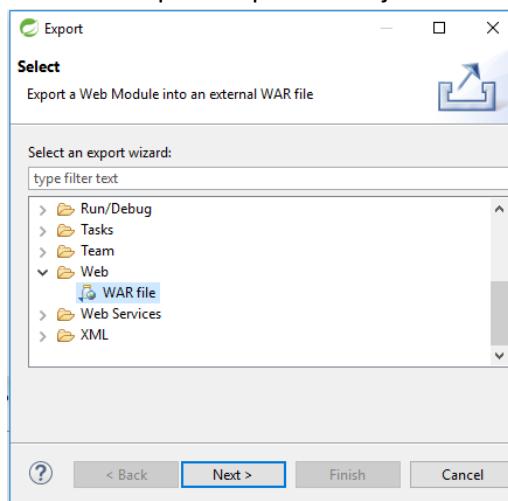


Figura 150 Exportación como fichero war paso 1

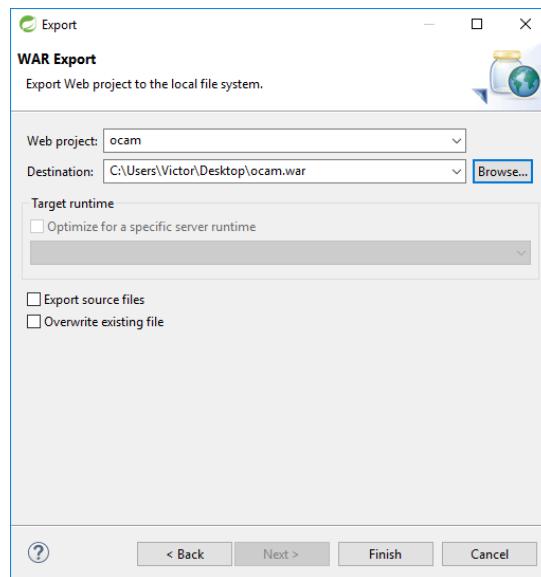


Figura 151 Exportación como fichero war – Paso 2

8.3 Manual de Usuario

En este apartado se va a explicar mediante capturas de pantalla como utilizar toda la funcionalidad que ofrece el sistema, tanto en la aplicación web como en la aplicación móvil.

8.3.1 Aplicación web

La aplicación web podemos encontrarla accesible en: <https://victorcastanogutierrez.github.io/OCAM-web/>. Cabe destacar, que al tratarse de un hosting gratuito donde se encuentra el servidor desplegado, es posible que tarde hasta un minuto en responder la primera vez, después lo hará con normalidad. Una vez la página haya cargado nos encontraremos aquí:

8.3.1.1 Acceso: login y registro

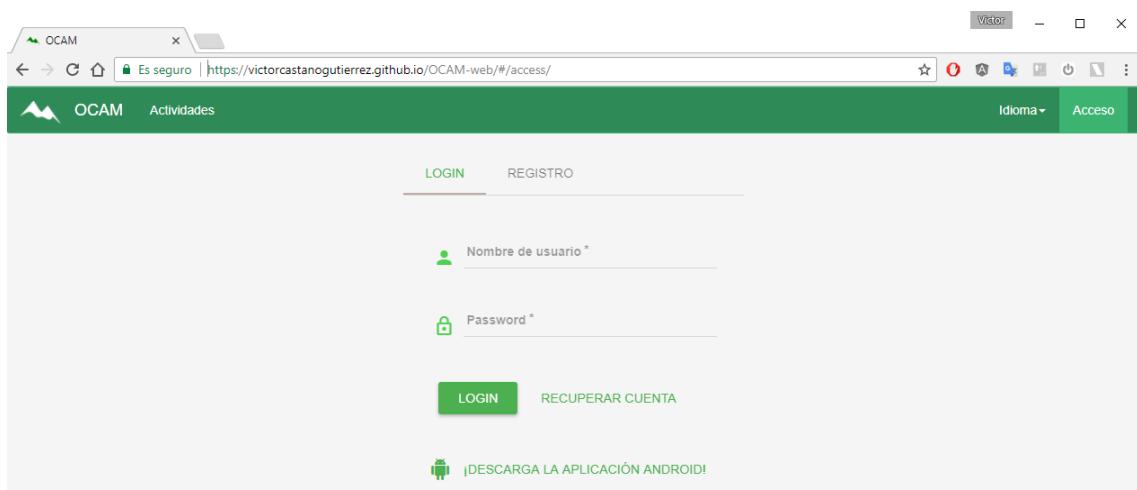


Figura 152 Login aplicación web

Lo primero que deberemos hacer es registrarnos en la aplicación. Para ello haremos click en la pestaña “Registro” del cuadro de acceso, lo cual nos mostrará el formulario para registrarnos en la aplicación.

Formulario de registro para OCAM. Los campos son:

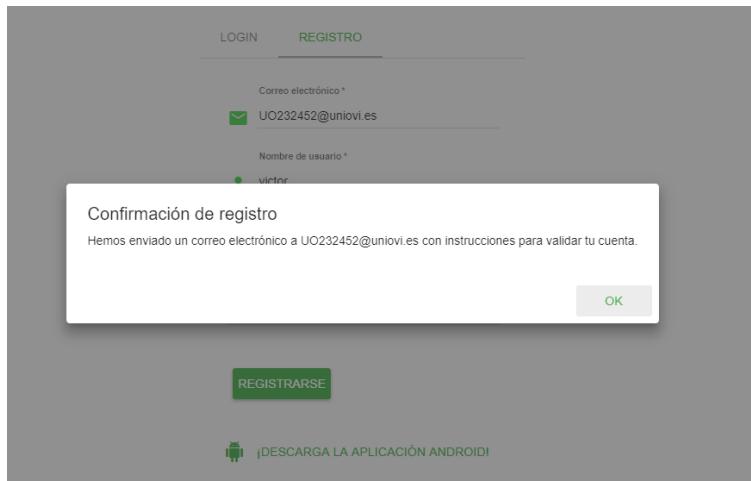
- Correo electrónico*: UO232452@uniovi.es
- Nombre de usuario*: victor
- Password*: ****
- Re password*: ****

Botón: REGISTRARSE

Botón: ¡DESCARGA LA APLICACIÓN ANDROID!

Figura 153 Formulario de registro

Debido al mismo motivo de estar desplegado el servidor en un hosting gratuito, no permite el envío de emails y por lo tanto está configurado para validar la cuenta de forma automática. De no ser así, al registrarnos habríamos obtenido el siguiente mensaje:

**Figura 154 Aviso de envío de email de confirmación**

Una vez el servidor nos envía el email, comprobamos en la dirección aportada y tendríamos el siguiente correo electrónico:

**Figura 155 Email de confirmación de cuenta**

Y accediendo al enlace que nos aporta el propio correo electrónico:

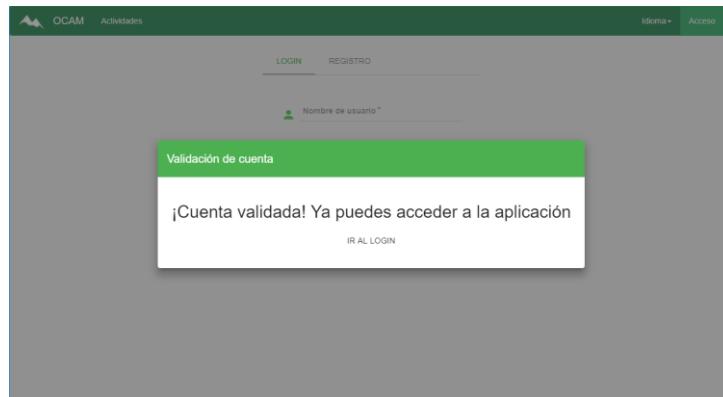


Figura 156 Aviso de cuenta validada

Al no poder enviar este correo electrónico, la validación de cuenta se hará de manera automática, tal y como informará la página web al hacer click en “Registrarse”, por tanto, no nos hará falta entrar en el correo electrónico.

Ya estamos registrados, podemos volver a la pestaña del login e introducir las credenciales que acabamos de utilizar para el registro.

8.3.1.2 *Listado de actividades*

A screenshot of the OCAM application's main dashboard. The title bar says 'Actividades' and 'Nueva actividad'. The top right shows a welcome message '¡Bienvenido victor!' and a user icon. The main area is titled 'Actividades pendientes'. Below it is a table with two rows of activity data. The columns are 'Fecha prevista' (Scheduled Date), 'Plazas máximas' (Max Seats), 'Descripción' (Description), and 'Estado' (Status). The first row has a green 'En curso' (In Progress) status, and the second row has a red 'Pendiente' (Pending) status. At the bottom right of the table is a red '+' button.

Figura 157 Listado de actividades pendientes página web

Estamos en la página principal de la aplicación, la cual también es accesible para un usuario no identificado. Desde aquí podemos ver el listado de actividades pendientes.

Además, por el hecho de estar identificados tenemos también acceso a las actividades realizadas. También por el mismo motivo podemos acceder al detalle de cualquier actividad haciendo click en cualquier punto de la fila en la tabla.

8.3.1.3 Detalle de actividad

Accedemos a una actividad que esté en curso, en el caso de la imagen “Actividad formulario usabilidad”. Al entrar en dicha actividad podemos ver todos los detalles de la misma: organizador, descripciones, fecha, plazas, guías... etc.

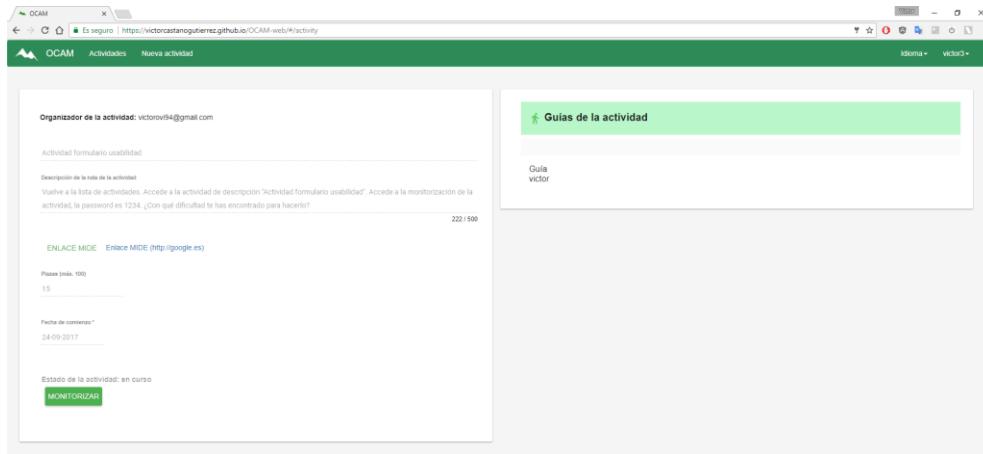


Figura 158 Detalle de una actividad en curso

En la parte inferior de la pantalla podemos ver el track de la ruta. Está dibujado a modo de polylinea sobre un mapa. Además, para aumentar el nivel de detalle, sobre el mapa se puede observar una rejilla que ilustra en cada vértice las coordenadas GPS, además de la distancia entre vértice y vértice en cada momento.

En la parte superior izquierda del mapa podemos cambiar de proveedor. Por defecto viene el mapa Raster, el cual solo obtiene datos del territorio en España. Si hacemos el suficiente zoom sobre el track de la ruta, podremos ver los puntos exactos por los que pasa.

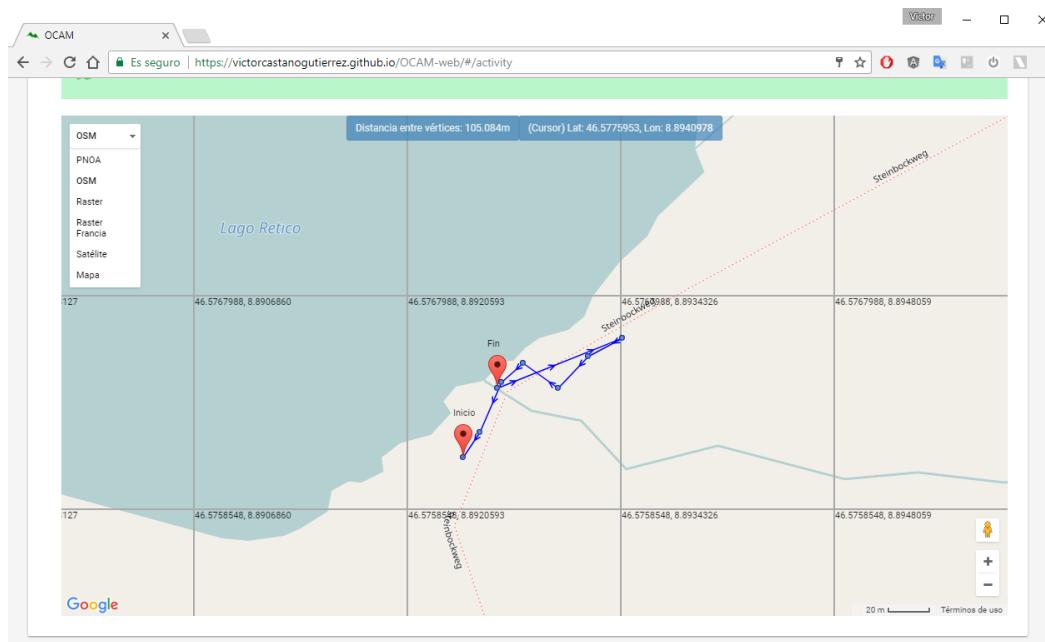


Figura 159 Track de la ruta

Por el hecho de estar la actividad en curso podemos acceder a la monitorización de la misma. Esto es el seguimiento continuo a cada uno de los participantes de la actividad. Para ello hacemos click en el botón de “MONITORIZAR”. Nos pedirá una password de acceso, en este caso para esta actividad en concreto es “1234”. La password la establece un guía al comenzar la actividad desde la aplicación móvil.

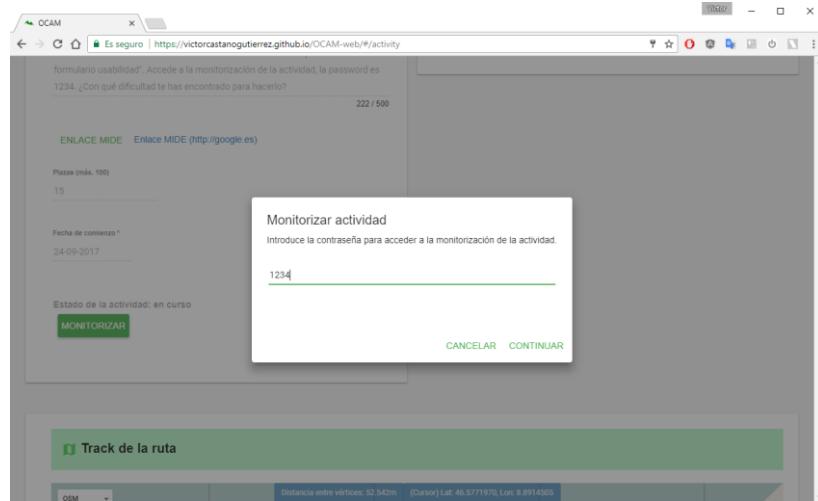


Figura 160 Password para monitorizar actividad

Desde esta pantalla podemos ver toda la información de todos los excursionistas. En este caso en concreto utilizado, hay dos excursionistas siendo monitorizados en la actividad.

8.3.1.4 Monitorización de actividad

A screenshot of the OCAM web application showing the 'Monitorización de excursionistas' (Monitoring of hikers) section. The table lists two hikers: 'formulario@gmail.com' and 'victorovi94@gmail.com'. For each hiker, there is an 'Email' checkbox, the date 'Último reporte' (Last report), and a 'MOSTRAR TRAYECTORIA' (Show trajectory) button with a green circular icon. The table has a header row with icons for sorting and filtering. At the bottom, there are pagination controls: 'Página: 1', 'Filas por página: Todos (2)', and links for '1 - 2 de 2', '<', '>', and '>>'.

Figura 161 Excursionistas monitorizados vista de monitorización

Podemos filtrar la lista por correo electrónico, recargar todos los datos o colapsarla. Colapsar la lista será de utilidad cuando sea demasiado grande verticalmente y se quiera tener de manera accesible el mapa de monitorización, situado debajo de dicha tabla. Para cada excursionista podemos ver la siguiente información:

- Correo electrónico

- Fecha y hora del último reporte enviado. En la figura anterior, uno de los dos excursionistas aún no ha enviado ningún reporte de posición.
- Acceso a mostrar/ocultar la trayectoria del excursionista sobre el mapa
- Recargar la información y reportes de solo ese excursionista.

Haciendo click sobre un excursionista de la tabla marcamos su check. Esto hará que se muestre un marcador sobre el mapa con la última posición conocida de dicho excursionista en la actividad. Además, podemos mostrar y ocultar el track de la ruta.

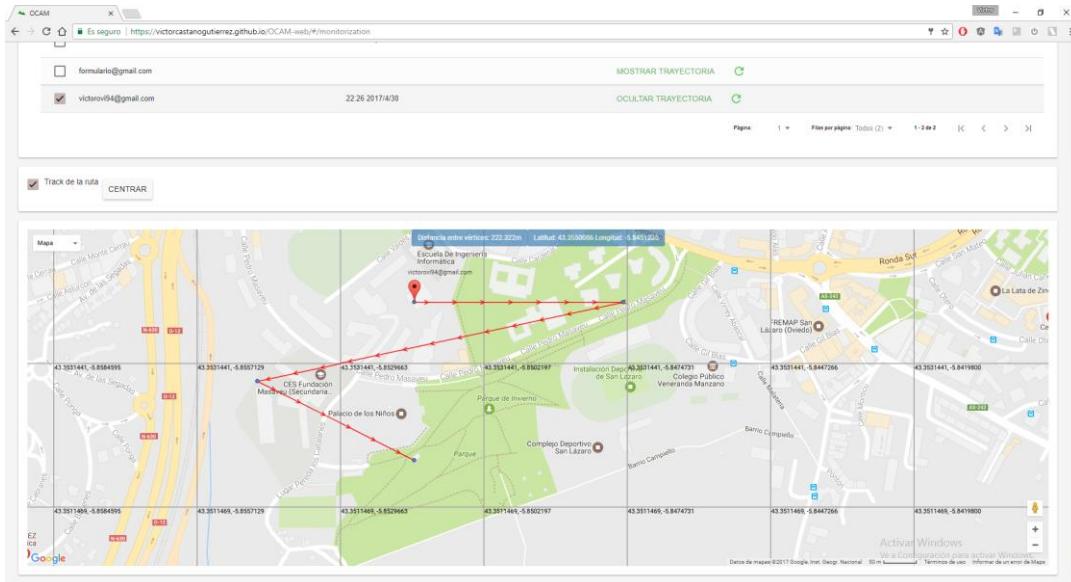


Figura 162 Monitorización de un excursionista

Como se puede ver en la anterior imagen, se ha seleccionado a un excursionista de la tabla. Se muestra por tanto en el mapa la trayectoria que ha seguido durante el transcurso de la actividad y su posición actual. En este caso, solamente ha enviado un total de cuatro reportes de posición.

De la misma manera que podemos ver y monitorizar actividades, se pueden ver los resúmenes de actividades ya realizadas. Para ello, volvemos al listado de actividades y filtramos por actividades realizadas utilizando el combobox encima de dicho listado.

8.3.1.5 Actividades realizadas

The screenshot shows a web browser window titled 'OCAM' with the URL <https://victorcastanogutierrez.github.io/OCAM-web/#/activities>. The page has a green header bar with the OCAM logo, the word 'Actividades', and a 'Nueva actividad' button. On the right, it shows 'Idioma' set to 'alejandro'. Below the header is a table titled 'Actividades' with columns: 'Fecha realización', 'Plazas máximas', and 'Descripción'. The table contains five rows of data:

Fecha realización	Plazas máximas	Descripción
2017-06-11	50	Prueba alejandro dia 11
2017-06-10	10	Prueba alejandro dia 10
2017-05-27	10	Prueba alejandro 27
2017-05-24	12	Prueba Alejandro 13
2017-05-11		Prueba 1 Alejandro

At the bottom right of the table area is a red '+' button.

Figura 163 Actividades realizadas

Un usuario solo puede ver las actividades en las que él mismo ha tomado parte, esto es, en las que en algún momento de la actividad participó en la misma por medio de la aplicación móvil.

Funciona de la misma manera que las actividades pendientes de realización, haciendo click sobre una de ellas entraremos al detalle. Cabe destacar que en este listado no se muestran estados porque todas las actividades que aquí aparezcan ya están acabadas y cerradas.

The screenshot shows a web browser window titled 'OCAM' with the URL <https://victorcastanogutierrez.github.io/OCAM-web/#/activity>. The page has a green header bar with the OCAM logo, the word 'Actividades', and a 'Nueva actividad' button. On the right, it shows 'Idioma' set to 'alejandro'. The main content area is divided into two sections. The left section displays activity details: 'Organizador de la actividad: victorovi94@gmail.com', 'Prueba alejandro dia 11', 'Descripción de la ruta de la actividad', 'Prueba dia 11', 'Plazas (máx. 100)', '50', 'Fecha de comienzo*', '20-07-2017', and a 'VER RESUMEN' button. The right section is titled 'Guías de la actividad' and lists 'Guía victor' and 'Guía alejandro'.

Figura 164 Detalle de actividad concluida

8.3.1.6 Resumen de actividad realizada

Dentro del detalle de la actividad vemos lo mismo que en una actividad que esté pendiente o en curso, con la salvedad de que ahora, en vez de a la monitorización para las que están en curso, podemos acceder al resumen. En este caso, para acceder al resumen no hará falta aportar la contraseña de la actividad, puesto que el que llega a este punto ha participado en la misma.

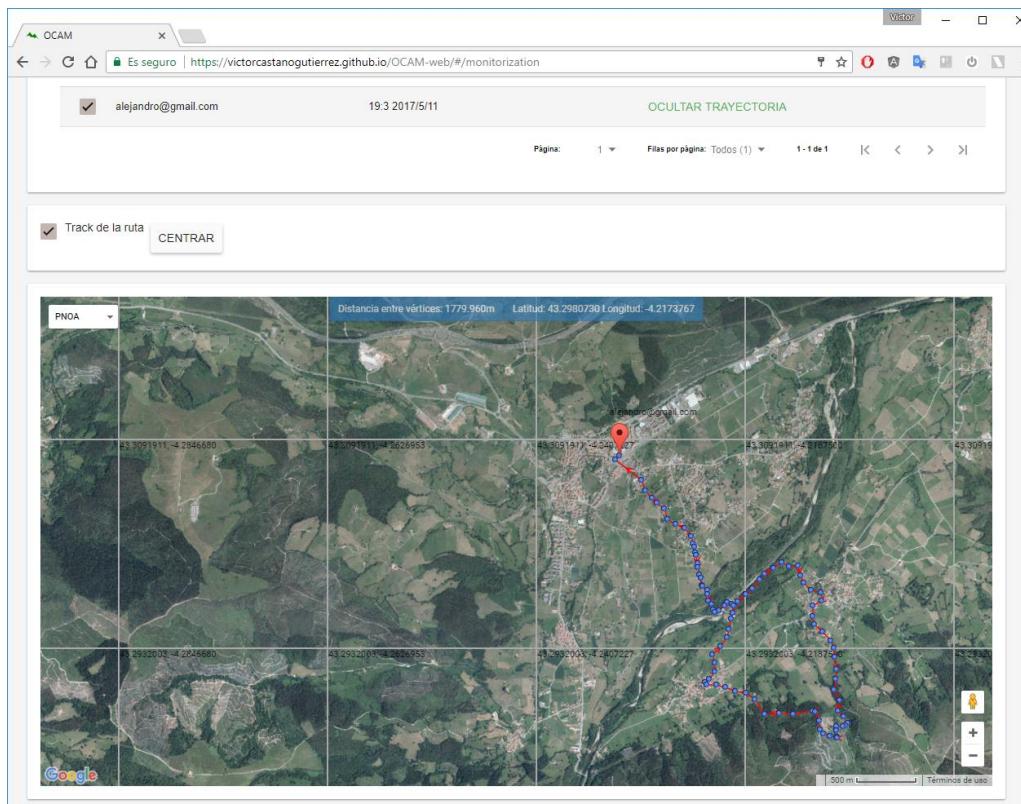


Figura 165 Resumen de monitorización

La funcionalidad de esta pantalla es similar a la de monitorización, solo que no se puede realizar actualizaciones de datos, puesto que al estar finalizada la actividad no tendría sentido.

Un usuario que no sea guía de la actividad, en el resumen solo le saldrá lo que ha hecho él/ella durante el transcurso. En caso de haber sido uno de los guías de la actividad, podrá ver en el resumen el detalle de todos los excursionistas que participaron.

8.3.1.7 Creación de actividades

Cualquier usuario identificado en el sistema puede crear actividades. Para acceder al formulario para crearlas se puede hacer de dos maneras: desde el listado a través del botón en la parte inferior del mismo con un “+” o bien desde la barra de navegación presente en todas las vistas a través de “Nueva actividad”.

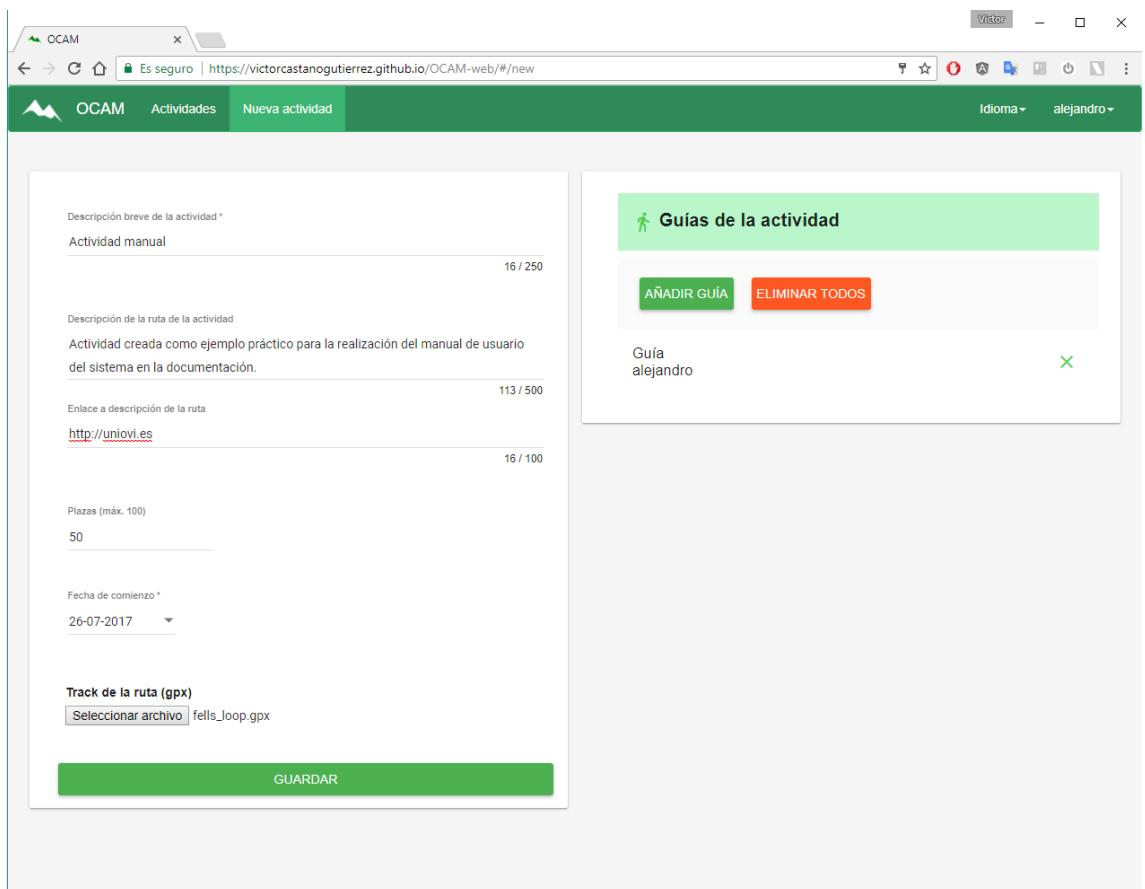


Figura 166 Creación de nueva actividad

Para crear una actividad será necesario de manera obligatoria aportar los siguientes datos sobre la misma:

- Descripción breve
- Descripción larga o enlace MIDE (o ambos). El enlace MIDE es un enlace que contiene la descripción de la ruta. Por tanto, para poder pasar la validación a la hora de crear la actividad se verificará que sea un enlace real.
- Fecha prevista de comienzo de la actividad
- Track de la ruta en formato “.gpx”
- Al menos un guía. Por defecto, el que crea la actividad figura añadido como guía de la misma, aunque puede eliminarse.

Para añadir un nuevo guía se deberá hacer click en “añadir guía” e introducir el nombre de usuario registrado en la aplicación de la persona deseada.

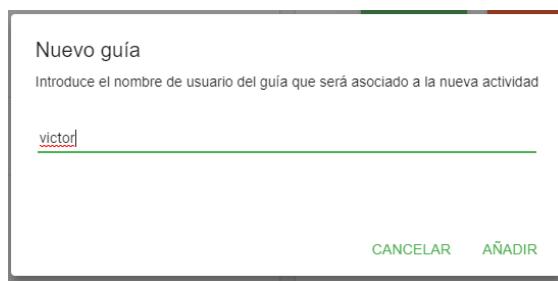


Figura 167 Nuevo guía en creación de actividad

Se puede además eliminar uno a uno cualquiera de los guías que se hayan añadido previamente o bien eliminarlos todos de golpe.

Si todos los datos introducidos para la actividad son correctos, podemos hacer click en “Guardar actividad”. Nos preguntará entonces si estamos seguros.

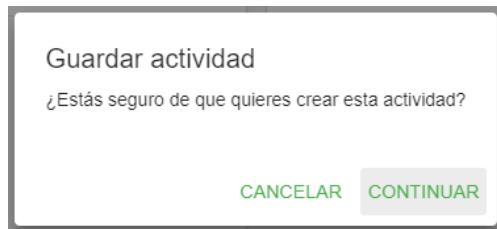
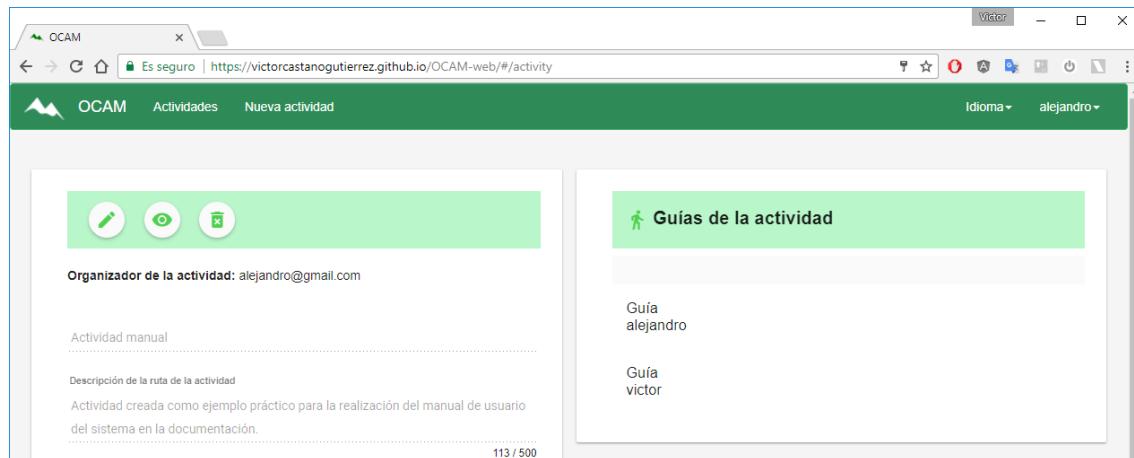


Figura 168 Guardar actividad

Si así es, continuamos y la actividad se creará. Redireccionará entonces al detalle de la actividad que acabamos de crear.

8.3.1.8 Edición de actividades

Desde el detalle de una actividad de la cual seamos el organizador podemos acceder a la edición de la misma o incluso a eliminarla a través de los accesos situados encima de la descripción de la actividad.



El primer botón nos permitirá editar la actividad, el segundo monitorizarla (en este caso, aunque lo intentásemos no sería posible puesto que no está iniciada) y el tercero y último eliminarla.

La edición de la actividad nos llevaría al mismo formulario utilizado para su creación. Se puede editar cualquier campo. En el caso de querer eliminarla nos saldría un previo aviso para confirmar la acción.

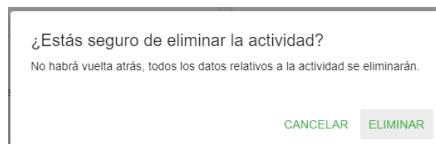


Figura 169 Confirmación de eliminación de actividad

8.3.1.9 *Mi cuenta*

Haciendo click en la barra de navegación sobre tu nombre de usuario puedes acceder a la ventana de “Mi cuenta”. Desde aquí se le permite a un usuario cambiar su contraseña (aportando la inicial) y eliminar la cuenta.

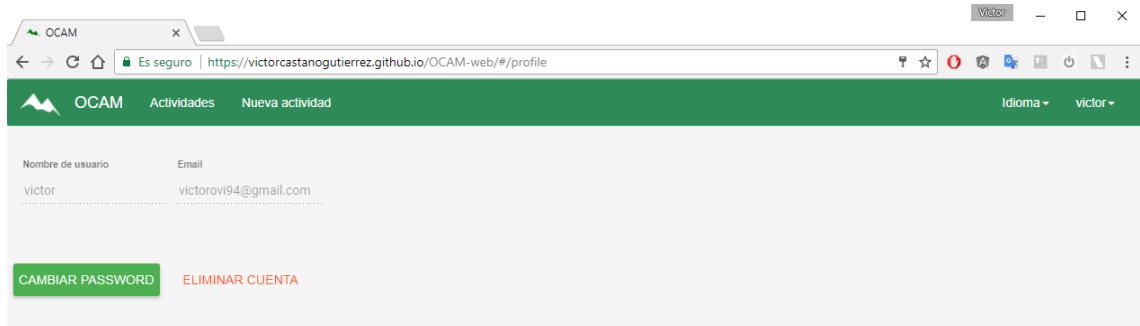


Figura 170 Mi cuenta

Figura 171 Cambiar password

8.3.2 Aplicación móvil

La aplicación móvil complementa la funcionalidad de la aplicación web, salvo en algunos aspectos como en el acceso.

8.3.2.1 *Acceso: login y registro*

De la misma manera que en la aplicación web, podemos iniciar sesión y registrarnos, mismas cuentas que luego podríamos utilizar también en la web.

**Figura 172 Login aplicación móvil****Figura 173 Registro aplicación móvil**

Mediante el check de “Recuérdame” del login indicamos a la aplicación nuestra intención de iniciar sesión de manera automática en las siguientes veces que la abramos.

Además, esto es necesario si queremos utilizar la actividad sin conexión a internet. Si marcamos el check de “Recuérdame” y accedemos a la aplicación sin conexión, entraremos de manera automática.

8.3.2.2 *Listado de actividades*

La primera vista a la que pasamos al iniciar sesión es a la lista de actividades. Aquí figurarán, de la misma manera que en la página web, aquellas que estén pendientes de realización o en curso en ese momento.

**Figura 174 Listado actividades sin conexión aplicación móvil**

Si entrásemos sin conexión saldría exactamente lo mismo, pero la aplicación nos avisaría del estado de la conectividad del dispositivo. Los datos que nos mostraría entonces serían los de la última vez que entramos en la aplicación con conexión a internet.

En la parte superior de la vista podremos filtrar las actividades del listado por fecha o bien recargar la información. Desplazando las actividades verticalmente hacia abajo también se recargan.



Figura 175 Filtrado lista actividades aplicación móvil



Figura 176 Recarga lista actividades aplicación móvil

Desde el listado de actividades la información que nos figura para cada actividad es la descripción breve, la fecha prevista de comienzo y el estado. Sin embargo, si accedemos a “Detalle” obtendremos más datos sin entrar en la propia actividad

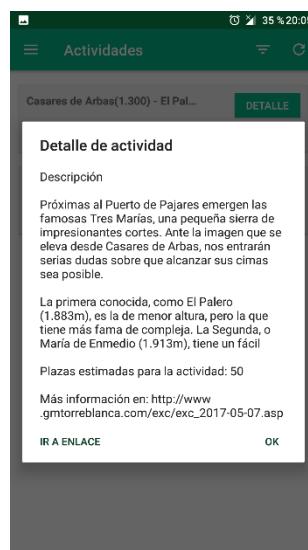


Figura 177 Detalle de una actividad – aplicación móvil

A través del botón “Ir a enlace” automáticamente se abrirá el navegador predeterminado del dispositivo con el enlace al detalle de la actividad, en caso de tenerlo.

8.3.2.3 *Detalle de una actividad*

Haciendo click en una de las actividades del listado accedemos al detalle de la misma. Desde el detalle podemos ver todos los datos de la actividad, incluido el track de la ruta situado en la esquina superior derecha.

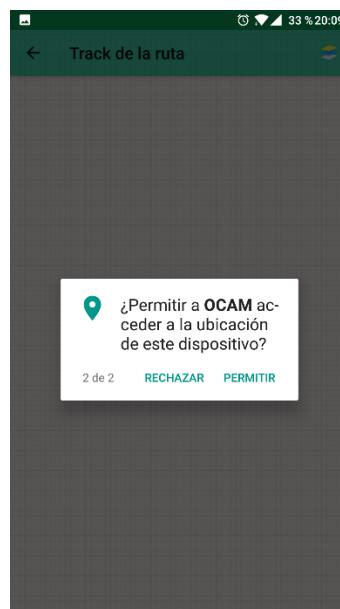


Figura 178 Solicitud de permisos aplicación android

La primera vez que se acceda a un mapa en la aplicación se nos solicitarán un total de dos permisos:

- Almacenamiento interno: los mapas se descargan de diferentes servidores. Para optimizar los tiempos de carga y la cantidad de datos a descargar, se cachean. Para cachear los datos de mapas ya descargados hace falta permisos de escritura en memoria. En ningún caso se llegan a escribir grandes cantidades.
- Ubicación: para muchas de las tareas que lleva a cabo la aplicación es necesaria la ubicación del dispositivo.

En la esquina superior derecha del mapa podremos cambiar entre tipos de mapas a mostrar entre diez opciones posibles. Por defecto, viene seleccionado OSM.

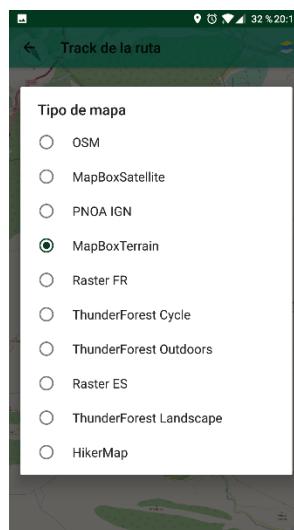


Figura 179 Tipos de mapa aplicación android

8.3.2.4 Gestión de actividades

Consideramos la gestión de actividades como al conjunto de acciones que cambian una actividad de un estado a otra interactuando para ello con los usuarios de la aplicación.

8.3.2.4.1 Comenzar actividad

Si somos guía de una actividad podemos llevar a cabo la gestión de la misma. Primero que nada, si la aplicación está en estado PENDIENTE, podremos darla por comenzada desde el detalle de la actividad. Para darla por comenzada nos preguntará la propia aplicación por una password. Dicha password será la que más adelante será necesaria para monitorizar la actividad y para que otros excursionistas se unan a ella a través de la aplicación.

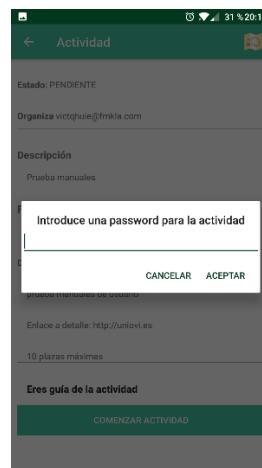


Figura 180 Introducción password para comenzar actividad

Una vez que como guía se da por iniciada la actividad, automáticamente el guía que la inició pasa a estar unido en ella y por tanto a ser monitorizado.

8.3.2.4.2 Cambiar password actividad

Una actividad comenzada tiene ya una password asociada. Esta password, si así fuese necesario, puede ser cambiada desde el detalle de la actividad por un guía de la misma.

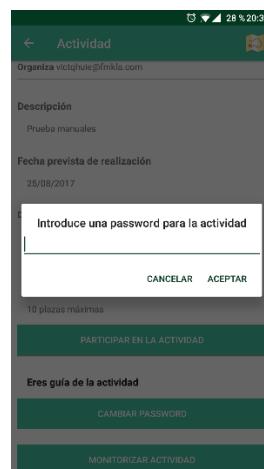


Figura 181 Cambiar password actividad

8.3.2.4.3 Unirse a actividad

Cualquier usuario registrado e identificado en la aplicación puede unirse a una actividad en curso. Para ello, a través del detalle de la misma debe acceder al botón para unirse a la actividad. La aplicación le pedirá entonces la password que el guía introdujo al darla por iniciada.

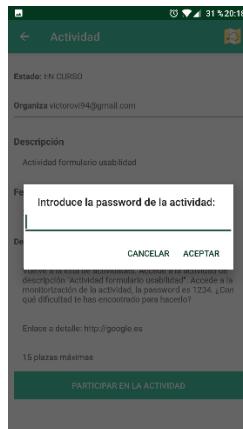


Figura 182 Password actividad unirse

8.3.2.4.4 Monitorizar actividad

Cualquier persona que se una a una actividad tendrá acceso desde el detalle de la misma a la monitorización. En la monitorización de una actividad se podrá ver sobre el mapa la posición actual de todos los excursionistas además del track de la ruta.

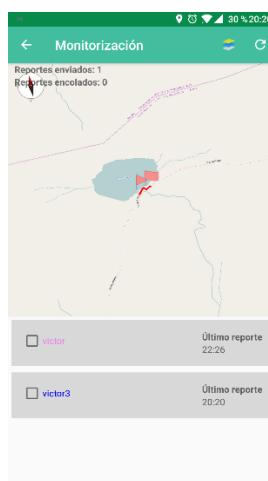


Figura 183 Monitorización actividad aplicación móvil

Al igual que con el track de la ruta, en esta vista también se puede cambiar el tipo de mapa. En la tabla de excursionistas salen todos aquellos unidos a la actividad con la hora en la que se emitió el último reporte de posición de cada uno de ellos.

Además, desde esta ventana se puede ver el resumen de reportes enviados y encolados. Los reportes enviados son todos aquellos que han llegado al servidor de manera satisfactoria. Los encolados son aquellos que por falta de conexión aún no ha sido posible

enviarse. En cuanto se recobre la conexión a internet, se enviarán con el siguiente reporte de posición.

Cada excursionista además tendrá asignado de manera aleatoria un color. Esto es para que al mostrar su posición en el mapa (marcando el check) el marcador que la ilustre llevará ese mismo botón. Así, si se muestran varios a la vez no se malinterpretan unos con otros.

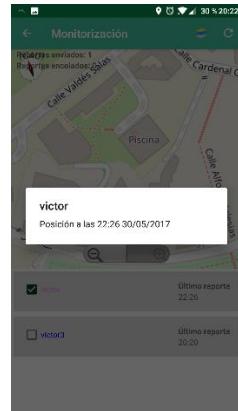


Figura 184 Posición excursionista mapa

Durante el desarrollo de la actividad, el dispositivo está recogiendo datos de posición de manera periódica, los reportes, que son enviados al servidor en caso de haber conexión a internet o encolados de forma local en caso de no tener conexión a internet.

La aplicación realizará esta tarea en segundo plano, esto quiere decir que, aunque se cierre y se bloquee el dispositivo todo funcionará con normalidad. Mandará una notificación cada vez que se envíe un reporte, o cada vez que no se pueda enviar por falta de conexión.



Figura 185 Notificación participante actividad aplicación móvil

8.3.2.4.5 Abandonar actividad

Cualquier excursionista unido a la actividad puede abandonarla a través del detalle de la misma. En el momento que abandone una actividad la monitorización se detendrá para ese dispositivo. Si así lo desease, se puede volver a unir en el futuro.

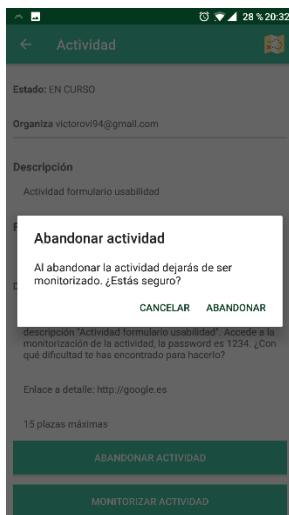


Figura 186 Confirmación abandono de actividad aplicación móvil

8.3.2.4.6 Cerrar actividad

De la misma manera que se inicia una actividad puede darse por concluida. Esto lo hará uno de los guías de la actividad. Al cerrarla pasa a estado realizada y todos aquellos dispositivos de excursionistas que estén siendo monitorizados dejan de estarlo. Es decir, todos los unidos a la actividad la abandonan al ser cerrada.



En la foto anterior se aprecia la notificación que le llega a todos los dispositivos unidos a la actividad que acaba de ser cerrada por uno de sus guías.

8.3.2.5 Navegación por la aplicación

A través del navigation drawer, accesible desde la lista de actividades en la esquina superior izquierda, se puede navegar por las otras funcionalidades de la aplicación.

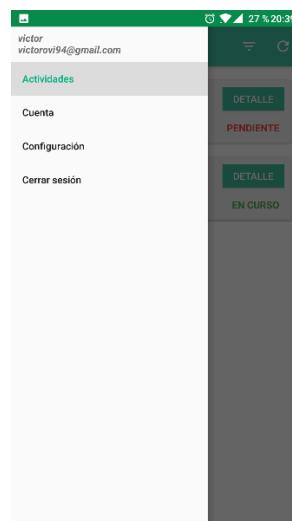


Figura 187 Menú de navegación aplicación móvil

8.3.2.6 Cuenta

Desde la vista de “Cuenta” se podrá ver tu nombre de usuario, correo electrónico y se tendrá la posibilidad de modificar la contraseña.

8.3.2.7 Configuración

Este es un aspecto importante para el uso de la aplicación. En la vista de configuración se puede alterar el valor numérico que luego será usado para la temporalidad de los reportes durante la monitorización.

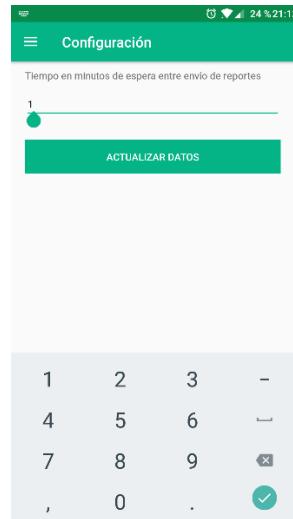


Figura 188 Configuración temporalidad reportes aplicación móvil

El número ahí especificado será en minutos y será el tiempo que transcurra entre reporte y reporte de posición. Cuanta mayor precisión a la hora de saber la posición de un excursionista, conviene que este número sea menor. El mínimo es 1 minuto y el máximo 360. Una vez introducido el valor deseado, haciendo click en “actualizar datos” se guardará la configuración. Esto es independiente de la conexión a internet, se guarda de manera local.

Si se pone un valor demasiado elevado y la actividad no es muy duradera, se corre la posibilidad incluso de que nunca llegue a enviarse ningún reporte de posición.

8.3.2.8 *Cerrar sesión*

En caso de cerrar sesión y tener la cuenta recordada se eliminaría y en sucesivos accesos a la aplicación volvería a preguntar por las credenciales.

8.3.2.9 *Acciones sin conexión*

Una de las características de la aplicación móvil es que puede funcionar sin conexión. Cuando se navega por la aplicación con conexión a internet se guarda de manera local en el dispositivo la información que se descarga del servidor.

En sucesivas entradas a la aplicación sin conexión a internet se mostraría lo último descargado del servidor.

Además, se pueden llevar a cabo diferentes acciones sin tener conexión. Se guardaría en la aplicación de manera local la acción realizada, los parámetros necesarios y se enviaría al servidor cuando se tuviera conexión. Posteriormente, se notifica al usuario de la realización correcta, o no, de dicha acción. Las acciones que se pueden completar sin conexión a internet son:

- Cerrar una actividad
- Comenzar una actividad
- Unirse a una actividad
- Abandonar una actividad

8.4 Manual del Programador

8.4.1 Servidor

Para poder desarrollar y probar el servidor será necesaria la instalación del entorno de desarrollo. En concreto, Eclipse, instalando las librerías necesarias podría ser válido para la tarea, pero se recomienda el uso de STS.

Para poder compilar y ejecutar el código del servidor será necesario establecer Java 1.8 por defecto en las propiedades del entorno > Java > Compiler.

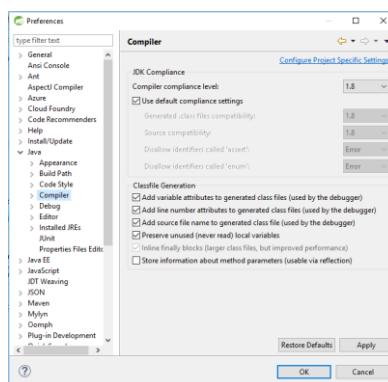


Figura 189 Java 1.8 compiler

8.4.2 Aplicación móvil

Para el desarrollo de la aplicación móvil se necesitará instalar Android Studio. Con cualquier versión superior a la 1.5 valdría para el desarrollo de la aplicación, aunque es altamente recomendable contar siempre con la más actualizada.

Para poder compilar las librerías de las que hace uso el proyecto se necesitará instalar Android 7.1.1 (Nougat) en el apartado de Android SDK de la configuración por defecto del Android SDK Manager.

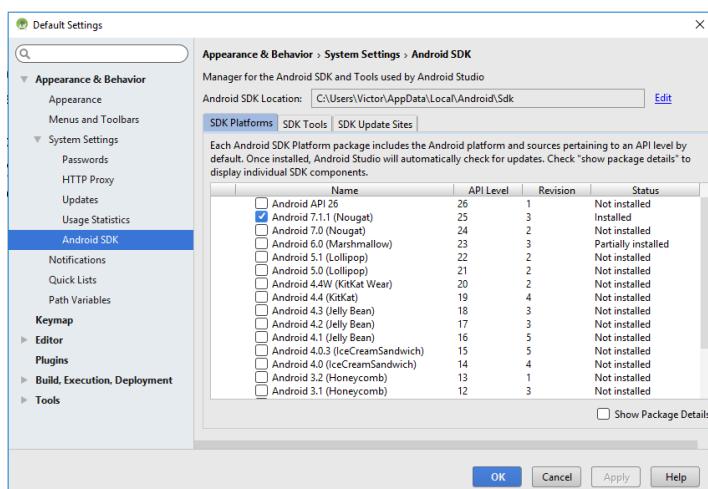


Figura 190 Android SDK Manager

8.4.3 Aplicación web

La edición del código de la aplicación web puede llevarse a cabo desde cualquier entorno a elección del propio programador.

Con el fin de ver en tiempo real los cambios que se realizan sobre los ficheros de código sin necesidad de recargar la página, es recomendable la instalación de browserSync.

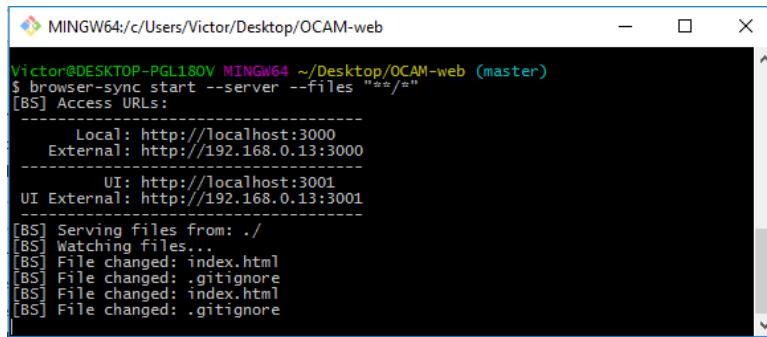
Para ello primero habrá que descargar NodeJS. Esto se puede hacer desde la página oficial de descargas seleccionando aquella que sea acorde a nuestro sistema operativo: <https://nodejs.org/es/download/>

Una vez descargado e instalado utilizamos el siguiente comando para instalar browserSync en nuestro ordenador:

```
npm install -g browser-sync
```

Una vez instalado podemos arrancarlo. Para ello nos situamos con la ventana de comandos en la carpeta donde esté el código fuente de la página web y mediante el siguiente comando se abrirá un servidor HTTP local en el puerto 3000 que escuchará por todos los cambios que se realicen sobre el código.

```
browser-sync start --server --files "*//*"
```



```
MINGW64:c/Users/Victor/Desktop/OCAM-web
Victor@DESKTOP-PGL180V MINGW64 ~/Desktop/OCAM-web (master)
$ browser-sync start --server --files "*//*"
[BS] Access URLs:
[BS]   Local: http://localhost:3000
[BS]   External: http://192.168.0.13:3000
[BS] 
[BS]   UI: http://localhost:3001
[BS]   UI External: http://192.168.0.13:3001
[BS] 
[BS] Serving files from: ../
[BS] Watching files...
[BS] File changed: index.html
[BS] File changed: .gitignore
[BS] File changed: index.html
[BS] File changed: .gitignore
```

Figura 191 Arrancando browser-sync en localhost:3000

Capítulo 9. Conclusiones y Ampliaciones

9.1 Conclusiones

9.1.1 Conclusiones técnicas

Se puede concluir en que **el proyecto se da por finalizado** en base a haber cumplido, por un lado, todos los **requisitos funcionales** y por otro los **no funcionales**.

Por medio de las matrices de trazabilidad con los casos de uso, se puede observar que todos están cubiertos por interfaces de usuario y posteriormente todos los requisitos cubiertos por dichos casos de uso. A su vez, de los casos de uso se extraen escenarios de los cuales se derivan las pruebas de aceptación. Al pasarse estas últimas con éxito se puede argumentar razonablemente que los requisitos funcionales se satisfacen.

A continuación, tanto para el subsistema aplicación web como aplicación móvil, se recoge la **medición de la respuesta** a cada uno de los requisitos no funcionales planteados en la fase de análisis del proyecto.

9.1.1.1 Aplicación web

- La conexión con el servidor, tal y como se explicó en el capítulo de diseño en 5.3, se realiza con el protocolo HTTPS además de llevar la información de usuario no sensible encriptada en un token JWT.
- La usabilidad ha sido garantizada tal y como se justificó en el capítulo de pruebas de usabilidad en 7.4 mediante encuestas a usuarios finales.
- Los navegadores en los que se puede utilizar la aplicación fueron probados en el capítulo de pruebas en 7.5.1.1 .
- La responsividad de la aplicación ha sido medida en el capítulo de pruebas, en 7.5.1.2 .
- De la misma manera que se explicó para la seguridad de las conexiones, ningún usuario puede acceder a información privilegiada de otros usuarios.
- La aplicación web ha sido internacionalizada en inglés y español.

9.1.1.2 Aplicación móvil

- De la misma manera que con la aplicación web, la conexión con el servidor, tal y como se explicó en el capítulo de diseño en 5.3, se realiza con el protocolo HTTPS además de llevar la información de usuario no sensible encriptada en un token JWT.
- La aplicación es ejecutable en cualquier versión de Android igual o superior a la 5.0.

- De la misma manera que con la aplicación web, la usabilidad ha sido garantizada tal y como se justificó en el capítulo de pruebas de usabilidad en 7.4 mediante encuestas a usuarios finales.
- Tal y como se expuso en el capítulo de implementación en 6.5.1, bajo un tiempo de trabajo de 120 minutos con reportes cada 60 segundos, la aplicación consume un 9% de batería.

9.1.2 Conclusiones académicas

En primer lugar, creo que fue un gran acierto la utilización de cliente totalmente desvinculado del servidor que consumiera los datos a través de interfaces REST. Creo que la manera en la que se pensó la arquitectura del sistema, como los clientes consumen, envían datos y se autentican permite que el proyecto sea totalmente escalable.

Para el desarrollo de la aplicación móvil, estaba claro en un principio que era necesario utilizar el patrón MVC con el fin principal de separar toda la lógica de las *activities* donde se presentaban los datos. La elección para ello del patrón MVP (derivado esencialmente del MVC) creo que suplió bien este requisito que nos planteamos desde el principio. Si bien es verdad que hay otros patrones para conseguir este mismo fin en Android, el haber utilizado MVP ha mejorado la mantenibilidad del código y la escalabilidad de la aplicación.

Quizás el único aspecto negativo con respecto a la aplicación móvil haya sido la compatibilidad con versiones de Android inferiores a la 5.0. Se intentó desde un principio mantener el requisito de funcionamiento en versiones como la 4.4 o incluso 4.0 pero debido a librerías que fue necesario utilizar para el desarrollo, esto no fue posible.

Para poder adaptar la aplicación móvil a cualquier tamaño de ventana se ha utilizado siempre tamaños fijos de proporción de resolución. Esto es posible gracias a la utilización de *weight* como parámetro en todas las *views* de los *layouts*.

Por último, con respecto a la mantenibilidad de los datos en local de la aplicación móvil, en un principio se intentó llevar a cabo esta tarea a través de bases de datos SQLite. Esto pronto llegamos a la conclusión de que acabaría siendo inmantenible debido a que a medida que crecía la aplicación era más y más complicado relacionar los datos. Para solucionar este problema se decidió utilizar la librería GreenDAO de la cual se ha podido obtener un rendimiento alto con unos buenos resultados.

9.1.3 Conclusiones personales

En un principio la aplicación no presentaba gran complejidad para su implementación, aunque finalmente estaba equivocado en este aspecto. Todo, a excepción de AngularJS había sido visto en la carrera, en mayor o menor medida.

El primer gran problema con el que me encontré fue con la autenticación del servicio REST. No era suficiente la utilización de HTTPS para garantizar la integridad de los datos por lo que fue necesaria la implementación de esta autenticación, no solo en el servidor donde fue más compleja la tarea, sino también preparar los clientes para su utilización.

Por otro lado, creo que fue un total acierto la realización del curso inicial de formación de AngularJS. Personalmente siempre he tenido muy claro que lo que me atraía era el frontend de las aplicaciones web, pero gracias a la realización de este curso me he dado cuenta de que la programación de clientes web en Javascript también puede ser apasionante.

Finalmente decir que antes de la realización de la aplicación móvil nunca pensé que fuera a ser la parte del proyecto que más esfuerzo me fuese a llevar, y así fue. La parte más compleja de la fase de implementación fue la persistencia de los datos en local en la aplicación móvil.

9.2 Ampliaciones

Existen diferentes ampliaciones que no fueron incluidas como requisitos del proyecto por falta de tiempo para llevarlos a cabo:

- Mensajería: sería interesante añadir la opción de poder enviar mensajes desde la monitorización de la actividad en la web a los excursionistas. Para ello un buen enfoque, aunque habría que estudiarlo bien, podría ser el uso de Firebase, más concretamente del sistema de notificaciones.
- Rol de administrador: la aplicación en realidad no necesita directamente de una figura de administrador, puesto que no hay demasiados aspectos que requieran de atención de una persona con dicho rol. No obstante, podría implementarse garantizándole todos los permisos en todos los aspectos de la aplicación web.
- Estadísticas: actualmente en el resumen de una actividad no se muestran estadísticas del desarrollo de la misma. Podría realizarse una mayor recogida de datos durante la ejecución de una actividad que luego facilitase la obtención final de estadísticas.

Capítulo 10. Referencias Bibliográficas

Javascript: The Good Parts – Douglas Crockford

Patrones de Diseño – Elementos de software orientado a objetos reutilizable – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides – Traducción César Fernández Acebal

Android Developers . (2017). Developer.android.com. Retrieved 27 June 2017, from <https://developer.android.com/index.html>

Leiva, A. (2014). MVP for Android: how to organize the presentation layer. Antonio Leiva. Retrieved 27 June 2017, from <https://antonioleiva.com/mvp-android/>

JWT Authentication Tutorial: An example using Spring Boot. (2016). Vladimir Stankovic. Retrieved 27 June 2017, from <http://www.svlada.com/jwt-token-authentication-with-spring-boot/>

Guía Breve de Accesibilidad Web. (2017). W3c.es. Retrieved 27 June 2017, from <http://w3c.es/Divulgacion/GuiasBreves/Accesibilidad>

La accesibilidad web en la legislación española < Artículos < codexexempla.org. (2017). Codexexempla.org. Retrieved 27 June 2017, from http://codexexempla.org/articulos/2008/legislacion_accesibilidad.php

Capítulo 11. Apéndices

11.1 Diario de trabajo

Se muestra a continuación el diario de trabajo al completo con un resumen de las tareas realizadas con la fecha y la carga de trabajo supuesta.

Fecha	Descripción	Duración (horas)	Horas acumuladas
25/10/2016	Reunión con el tutor del trabajo	1	1
26/10/2016	Introducción de documentación	3	4
27/10/2016	Introducción de documentación	2	6
28/10/2016	Corrección de la justificación en la introducción	2	8
29/10/2016	Evaluación de posibles alternativas y soluciones a problemas comunes	10	18
30/10/2016	Corrección de la introducción	2	20
01/11/2016	Formación: semana 1 AngularJS	5,5	25,5
02/11/2016	Formación: semana 2 AngularJS	3	28,5
03/11/2016	Formación: semana 2 AngularJS	2,5	31
04/11/2016	Formación: semana 2 AngularJS	4	35
05/11/2016	Formación: semana 2 AngularJS	2,5	37,5
11/11/2016	Formación: semana 3 AngularJS	3	40,5
12/11/2016	Formación: semana 3 AngularJS	5	45,5
13/11/2016	Formación: semana 3 AngularJS	3,5	49
15/11/2016	Formación: semana 4 AngularJS	3	52
18/11/2016	Formación: semana 4 AngularJS	1,5	53,5
19/11/2016	Formación: semana 4 AngularJS	2	55,5
20/11/2016	Formación: semana 4 AngularJS	3	58,5
22/11/2016	Formación: semana 5 AngularJS	5	63,5
23/11/2016	Formación: semana 5 AngularJS	4	67,5
24/11/2016	Formación: semana 5 AngularJS	0,5	68
25/11/2016	Formación: semana 5 AngularJS	6	74
26/11/2016	Formación: semana 5 AngularJS	4	78
01/12/2016	Reunión con el tutor del trabajo	0,75	78,75
08/12/2016	Reescribir solución propuesta	4	82,75
09/12/2016	Requisitos subsistema página web	6	88,75
22/12/2016	Requisitos subsistema aplicación móvil	5	93,75
26/12/2016	Mapa de navegación aplicación web	2	95,75
26/12/2016	Mapa de navegación aplicación web y prototipo de login	2	97,75
29/12/2016	Prototipos de pantalla aplicación web	1,5	99,25
30/12/2016	Prototipos de pantalla aplicación web	0,5	99,75
18/01/2017	Prototipos de pantalla aplicación móvil	1,5	101,25
24/01/2017	Reunión con el tutor del trabajo	1,5	102,75

25/01/2017	Segunda versión de prototipos móvil y web	2	104,75
26/01/2017	Ampliación de los requisitos funcionales con la nueva funcionalidad de la aplicación	1,5	106,25
26/01/2017	Realización de casos de uso alto nivel: aplicación web y móvil	2,5	108,75
28/01/2017	Prototipo de monitorización en navegador	4	112,75
29/01/2017	Prototipo de monitorización en navegador	3,5	116,25
01/01/2017	Casos de uso a nivel detallado	6	122,25
02/02/2017	Casos de uso a nivel detallado	6	128,25
04/02/2017	Casos de uso a nivel detallado, diagramas de actividad para cada caso de uso	4	132,25
07/02/2017	Retocar casos de uso e introducción del caso de uso de Iniciar sesión, retocar diagramas actividad y comenzar con pruebas de aceptación	5	137,25
08/02/2017	Realización de pruebas de aceptación subsistema web	10	147,25
09/02/2017	Realización de pruebas sobre tecnologías de mapas en android para la preparación de la fase de diseño	6	153,25
10/02/2017	Realización de pruebas sobre una aplicación Android en geoposicionamiento	4	157,25
11/02/2017	Realización de pruebas con Android en programación de tareas en segundo plano	5	162,25
12/02/2017	Diseño de servicios de negocio	6	168,25
13/02/2017	Diseño de servicios de negocio	7	175,25
14/02/2017	Inclusión en el capítulo de análisis de la documentación el análisis de la primera versión de los prototipos.	6	181,25
14/02/2017	Realización de un primer modelo de dominio	3	184,25
16/02/2017	Reunión con el tutor del trabajo: revisión general de la fase de análisis y diseño y planificación de la fase de implementación	1,5	185,75
17/02/2017	Creación del servidor. Programación de test unitarios básicos para la capa de servicio. Programación del modelo de dominio y primeros servicios.	5	190,75
18/02/2017	Programación usando TDD de los servicios del servidor	6	196,75
19/02/2017	Programación del test y del servicio ActivityService	5	201,75
20/02/2017	Programación del test y del servicio ActivityService	6	207,75
24/02/2017	Programación de la autorización de servicio REST en el servidor basada en token	4	211,75
25/02/2017	Programación de la autorización de servicio REST en el servidor basada en token	2	213,75
26/02/2017	Generación de la keystore para cifrado de las comunicaciones (HTTPS).	3	216,75
26/02/2017	Encriptación de las contraseñas guardadas en base de dato bajo algoritmo MD5. Actualización de los test con el nuevo funcionamiento de las password	2,5	219,25
01/03/2017	Programación del controlador REST de Hiker	1,5	220,75
02/03/2017	Comienzo de programación de la página web. Creado el proyecto y módulos de la aplicación. Creada página de login y petición POST al servidor	2	222,75
03/03/2017	Programación de la página de login	4	226,75
04/03/2017	Refactorización del código de la página de login en	2	228,75

	componentes. Inclusión de la libería gráfica material.angularjs al proyecto y diseño de la vista de login y registro. Programación del registro en la aplicación. Pequeños cambios en el controlador REST hiker del servidor. Inclusión de un nuevo entry point REST para comprobar si un Hiker existe o no		
05/03/2017	Refactorización del código de errores en página de login y registro. Refactorización en dos servicios diferentes el login y los accesos a ws de Hiker	2	230,75
06/03/2017	Realización de configuración para CORS en el lado servidor	3	233,75
07/03/2017	Programación parte pública de la página web	1,5	235,25
08/03/2017	Programación de la lista infinita en la página de actividades pendientes	2,5	237,75
09/03/2017	Programación de lista principal paginada en angular material	4	241,75
10/03/2017	Lista principal	3	244,75
11/03/2017	Página de perfil	1	245,75
13/03/2017	Cambio de contraseña	1	246,75
14/03/2017	Programación web	2	248,75
15/03/2017	Solución de errores en página de acceso. Header responsivo	2	250,75
16/03/2017	Estudio para integración con angular google maps. Página de actividad. Solución de errores en lista de páginas	2	252,75
17/03/2017	Cambios en seguridad CORS servidor	3	255,75
21/03/2017	Cambios visuales en lista de actividades. Mapa en página de actividad	2	257,75
24/03/2017	Refactorización en componentes de la vista de actividad. Rediseñada vista ahora usando md-card para cada uno de los módulos. Implementada funcionalidad de guías	5	262,75
25/03/2017	Corrección de errores en la selección de guías para nueva actividad. Implementada edición de actividades. Cambios en la información que retorna el servidor con el token. Implementada funcionalidad para eliminar actividades del sistema (borrado lógico)	3	265,75
26/03/2017	Comprobación de password para monitorización	2	267,75
27/03/2017	Página de monitorización. Cambios en servicios. Añadida tabla con Hikers y posibilidad de filtro	3,5	271,25
28/03/2017	Mapa en página de monitorización con track de la ruta	2	273,25
29/03/2017	Añadida posibilidad de ver/ocultar última posición de cada Hiker en el mapa de monitorización. Además mostrar/ocultar/actualizar la trayectoria seguida durante toda la actividad para cada uno. Añadido envío de emails desde el servidor para confirmación de crear cuentas	3	276,25
30/03/2017	Mostrar y ocultar la trayectoria completa de un hiker. Validación de cuentas por medio de correo electrónico	3	279,25
31/03/2017	Reunión con el tutor del trabajo. Revisión de los avances en la página web y el servidor	1,5	280,75
31/03/2017	Cambios en campos opcionales en la creación de una nueva actividad. Despliegue de la aplicación en Heroku creando un perfil para PostgreSQL	1	281,75

01/03/2017	Introducción de diferentes tipos de mapa al mapa original de angular google maps	1,5	283,25
02/04/2017	Aplicación móvil. Creación del proyecto y repositorio. Login en la aplicación. Programación de clases genéricas para peticiones HTTP.	3	286,25
02/04/2017	Nueva API en servidor para comprobar si un hiker existe o no pasándole el email	2	288,25
02/04/2017	Fix en la resposividad de la aplicación en navegadores de dispositivos móviles	2,5	290,75
03/04/2017	Renovación del Token de forma automática al hacer login en la aplicación.	2,5	293,25
04/04/2017	API listado de actividades con identificación. Botón de refresco en la toolbar de la actividad principal. Lista de actividades en actividad principal. Cierre de sesión. Correcciones en el fallo de inicio de sesión	4,5	297,75
05/04/2017	Refactorizadas actividades a fragments incluidas en el navigation drawer principal de la aplicación. Correcciones en los menús	3	300,75
06/04/2017	Implementado listener para entrar a través del resumen de una actividad a su actividad. Implementada lógica para mostrar los datos de la actividad	2,5	303,25
07/04/2017	Implementado código para parsear contenido GPX en posiciones GPS en Android. Implementada visualización del track de la ruta en la actividad del detalle de una Actividad.	2	305,25
08/04/2017	Correcciones visuales en la vista de una actividad. Añadidos más datos recogidos al hacer login. Implementada lógica para iniciar una actividad. Editada carga de track para realizarse solo en el momento en el que se solicita explícitamente en la aplicación.	4,5	309,75
08/04/2017	Añadido servicio y API para iniciar una actividad. Corregido nombre de campo erroneo en respuesta HTTP. Añadida información de token, usuario y refreshToken al ampliar la duración de un token.	2	311,75
09/04/2017	Implementada actividad de monitorización con la lista de hikers monitorizados. Implementados botones para mostrar en el mapa última posición conocida de cada Hiker siendo monitorizado (mediante una lista). Implementada lógica para unirse a una actividad. Implementada lógica para cambiar la password de una actividad. Solucionado bug al enviar los datos del token contra la API de refresco	5	316,75
09/04/2017	Añadidas comprobaciones adicionales en el servicio de negocio para unirse a una actividad. Enviados datos de error con la respuesta 4XX de la API	2	318,75
10/04/2017	Implementado registro en la aplicación. Eliminadas progressBar por elemento de recarga. Implementada funcionalidad para refrescar los datos de los hikers en la actividad de monitorización.	4	322,75
10/04/2017	Implementado nueva API para obtener el listado de Hikers de una actividad	2	324,75
11/04/2017	Implementada funcionalidad para cerrar una actividad que esté en curso. Solucionado error al cerrar la actividad	3	327,75

	de registro.		
11/04/2017	Implementada API y servicio de negocio para cerrar una actividad	0,5	328,25
12/04/2017	Implementada API y servicio de negocio para guardar un report	1,5	329,75
12/04/2017	Envío de reportes desde la aplicación móvil estando conectado a internet	2	331,75
13/04/2017	Formación en Greenrobot (green DAO) para Android. Testing de anotaciones y generación de código con proyectos auxiliares.	6	337,75
13/04/2017	Inclusión de greenDAO en la aplicación y anotación de clases del modelo. Implementada lógica para identificar estado de conexión del dispositivo. Implementadas estrategias para guardar reporte en base de datos local cuando no haya conexión	4	341,75
14/04/2017	Volcado de datos recogidos sin conexión a peticiones POST para guardar todos los reportes. Investigación y corrección de diferentes bugs que bloqueaban el broadcast receiver cuando la aplicación había sido cerrada.	3	344,75
15/04/2017	Añadidos códigos de error en la autenticación en el servidor. Modificado el tiempo de vida en la generación de tokens. Cambios en la API de save report. Ahora la fecha la obtiene del DTO y no la genera en el momento.	1,5	346,25
15/04/2017	Añadido filtro por fechas en listado de actividades Android. Solucionados errores en el envío de reportes. Solucionados errores en la guarda del token	3,5	349,75
16/04/2017	Añadido punto de acceso a la API para eliminar un hiker de una actividad. Modificado punto de acceso para unirse a una actividad, añadido el parámetro password. Adaptados tests a nueva funcionalidad	1,5	351,25
16/04/2017	Funcionalidad en Android para abandonar una actividad. Funcionalidad para poder unirse a una actividad por medio de la password	3	354,25
17/04/2017	Investigación del mejor método para obtener la localización en Android. Cambio de Broadcast receiver a service. Realización de cambios visuales: estilo en actividades de login registro y lista de actividades	3	357,25
18/04/2017	Combinación de LocationUpdates (Service) con el antiguo servicio de Broadcast receiver	3	360,25
19/04/2017	Login en la aplicación móvil sin conexión a internet. Guardada más información en SharedPreferences. Cambios visuales en vista de actividad, toolbars y monitorización. Añadida advertencia de configuración no apta en GPS	3	363,25
20/04/2017	Guardado de actividades en local y volcado de esos datos en accesos a la aplicación sin conexión a internet. Implementación del botón y funcionalidad del mismo de detalle en cada actividad de la lista de actividades.	2,5	365,75
20/04/2017	Cambiado servicio para iniciar una actividad, ahora el único incluido como participante es el que realiza la petición a la API	2	367,75

21/04/2017	Solucionados errores en validaciones del servidor. Guardado de hikers en almacenamiento local al unirse a una actividad	2,5	370,25
22/04/2017	Cambiada forma de crear y eliminar los reportes pendientes de envío. Extraída funcionalidad a otras clases. Funcionalidad de iniciar una actividad sin conexión a internet. Correcciones visuales al login (logo). Funcionalidad para unirse a una actividad sin conexión a internet. Volcado de las acciones pendientes una vez recuperada la conexión. guardado del estado de unido en almacenamiento local al unirse a una actividad.	4	374,25
23/04/2017	Añadidos datos del hiker logueado al Nav drawer. Eliminados recursos no utilizados en el proyecto Android. Funcionalidad para abandonar una actividad sin conexión a inernet. Funcionalidad para cerrar una actividad sin conexión a internet. Acceso sin conexión a la actividad de monitorización por medio del volcado de datos locales	3	377,25
23/04/2017	Añadido un nuevo servicio y punto a través de la API para obtener las actividades de un Hiker en las que haya participado.	1,5	378,75
25/04/2017	Solucionado error en el envío de reportes desde Android estando conectado a internet. Instalación y configuración de Fabric en la aplicación	3	381,75
25/04/2017	Añadido nuevo servicio en servidor para obtener el último reporte de un hiker en una actiividad con acceso también a través de la API REST. Modificado el servicio para obtener los reportes de un hiker, ahora buscado por email. Adaptados los test a los cambios	3	384,75
25/04/2017	Nueva vista en la web para ver el resumen de una actividad cerrada. Filtro por actividades hechas y pendientes en la lista de actividades principal. Corrección de errores.	3	387,75
26/04/2017	Solucionados errores en la web con los tipos de mapa	4	391,75
27/04/2017	Reunión con el tutor del trabajo	1,5	393,25
27/04/2017	Añadida nueva fecha a la entidad Actividad para almacenar la fecha en la que se inició. De esta manera en la página web se distingue en el listado pendiente con la fecha prevista y en el listado realizadas con la fecha en la que se inició la actividad. Solucionado problema con la valdación del enlace MIDE	2	395,25
27/04/2017	Cambiado color y ancho de las polylíneas en la web. Cambiado a mapa por defecto Raster. Implementado movimiento del mapa a la hora de mostrar la posición de un hiker o su trayectoria. Al guía de una actividad se le muestran todos los participantes de la misma en el resumen de una que ya esté concluida. Cambiada ordenación, ahora por fechas, en las tablas de actividades.	2	397,25
28/04/2017	Añadido grid sobre los mapas de la web con información de las coordenadas en cada vértice. Calculada y mostrada distancia entre vértices del grid del mapa. Mostrado de coordenadas sobre las que se encuentra el cursor en cada momento. Implementación de puntos que forman las	2	399,25

	polylineas (de manera visual)		
29/04/2017	Añadida columna de estado en la lista de actividades de la web. Funcionalidad para colapsar el card con la información de los excursionistas en la página de monitorización. Implementada funcionalidad para recuperar la cuenta desde la página web en el formulario de login. Implementada funcionalidad para dar de baja una cuenta desde la página de perfil de la web.	4,5	403,75
29/04/2017	Añadido servicio de negocio que envía un mail para recuperar una password. Punto de acceso también a través de la API REST. Servicio y punto de acceso a través de la API REST para eliminar un hiker (borrado lógico)	2	405,75
30/04/2017	Implementada funcionalidad en Android para cambiar la password de la cuenta desde la página del perfil	2	407,75
01/05/2017	Solucionado error que truncaba ciertos parámetros en URL en la API REST. Añadido nuevo servicio para obtener los reportes de una actividad.	1,5	409,25
01/05/2017	Añadida notificación con la cantidad de reportes encolados en Android, en caso de haberlos. Añadidos indicadores a la actividad de monitorización con las cifras de reportes enviados y reportes encolados en ese momento. Añadidos y asignados a hikers colores aleatorios en la página de monitorización. Cambiado el color de los makers sobre el mapa mostrando los asignados previamente.	3	412,25
02/05/2017	Aumentados tamaños de string en dos atributos de Activity. Solucionado error en una query que afectaba al entorno en PostgreSQL	2	414,25
03/05/2017	Solucionado error al editar y eliminar una actividad en el servidor.	3	417,25
03/05/2017	Solucionados errores de validación de mide en la página web. Solucionados errores con la desaparición de los puntos que forman las polylineas.	2	419,25
04/05/2017	Solucionados errores en la actualización de los guías de una actividad. Incluido nuevo listener HTTP para manejar diferentes errores en respuestas del servidor. Bloqueado el cambio de orientación a horizontal de las actividades en Android	3	422,25
08/05/2017	Análisis en documentación: reescritos los requisitos funcionales	7	429,25
09/05/2017	Análisis en documentación: añadidos casos de uso restantes. Actualizadas referencias a requisitos desde casos de uso	6	435,25
10/05/2017	Cambiado a double los tipos de dato en servidor y aplicación móvil de la latitud y longitud	3,5	438,75
17/05/2017	Cambiada API de aplicación móvil para mapas a OSMdroid. Cambiado track de la ruta a nueva API	2,5	441,25
18/05/2017	Cambiada actividad de monitorización a nueva API OSMdroid	2	443,25
21/05/2017	Solucionados errores sin conexión con los mapas. Añadidos puntos de hikers y polylineas sobre mapas con nueva API	5	448,25

24/05/2017	Reunión con el tutor del TFG	1,5	449,75
28/05/2017	Añadida vista de configuración para configurar el tiempo entre reportes. Refactorización de código de Settings	4	453,75
28/05/2017	Internacionalización en aplicación web. Internacionalizadas páginas de login y registro	3,5	457,25
29/05/2017	Documentación: Capítulo 1	5	462,25
30/05/2017	Corrección de errores en servidor y web al eliminar una actividad. Corrección de errores con la internacionalización. Capítulo 2 documentación	6	468,25
31/05/2017	Solucion de bug en internacionalización al cambiar de idioma sucesivas veces en la misma visata. Documentación capítulos dos y tres	5	473,25
01/05/2017	Documentación: Capítulo 3. Internacionalización vista de actividad y nueva actividad	4	477,25
02/05/2017	Documentación. Fixes en página web	5	482,25
03/05/2017	Documentación: análisis	5	487,25
04/05/2017	Documentación: análisis	4	491,25
05/05/2017	Documentación: análisis	5	496,25
06/05/2017	Documentación: análisis y diseño. Formulario para android	7	503,25
07/05/2017	Documentación: diseño	6	509,25
08/05/2017	Documentación: diseño	6	515,25
09/05/2017	Documentación: diseño	7	522,25
10/05/2017	Documentación: implementación	6	528,25
11/05/2017	Documentación: implementación	6,5	534,75
12/05/2017	Pruebas	7	541,75
13/05/2017	Pruebas	5	546,75
14/05/2017	Pruebas	6	552,75
15/05/2017	Pruebas	5	557,75
16/05/2017	Documentación: pruebas	7	564,75
17/05/2017	Documentación: manuales	7	571,75
18/05/2017	Documentación: manuales	6	577,75
19/05/2017	Reunión con el tutor	1	578,75
20/05/2017	Documentación: conclusiones	2,5	581,25
21/05/2017	Revisión de documentación	5	586,25
22/05/2017	Revisión de documentación	4	590,25
23/05/2017	Revisión de documentación	4,5	594,75
24/05/2017	Revisión de documentación	3	597,75
25/05/2017	Revisión de documentación	5	602,75

Tabla 22 Diario de trabajo

11.2 Contenidos entregados

En este apéndice se describen los contenidos que han sido subidos al Sistema de Gestión de Trabajos Fin de Estudios.

11.2.1 Estructura de la carpeta entregada

- **Carpeta OCAM:** contiene el proyecto JAVA del servidor. Dentro de esa carpeta podemos encontrar:
 - o Certificado HTTPS **keystore.p12**.
 - o Carpeta **doc** con todo el javadoc del proyecto.
 - o Carpeta **src** contiene el código fuente.
- **Fichero ocam.war:** servidor exportado
- **Carpeta OCAM-web:** contiene el proyecto de la página web.
 - o Carpeta **src** contiene el código fuente.
 - o Fichero **index.html** punto de acceso a la aplicación web.
 - o Carpeta **spec** contiene el código fuente de las pruebas de comportamiento.
 - o Fichero **SpecRunner.html** punto de acceso a las pruebas de comportamiento de la página web.
 - o Fichero **ejemplo.gpx**: contiene un track a modo de ejemplo por si fuese necesaria su utilización para la realización de pruebas.
- **Carpeta OCAM-android:** contiene el proyecto de la aplicación móvil.
 - o Carpeta **app**: contiene el código fuente de la página web
- **Fichero app-release.apk:** última versión de la aplicación Android.
- **Fichero OCAM.vpp:** proyecto Visual Paradigm con todos los diagramas utilizados en la documentación del proyecto.
- **Fichero OCAM.pdf:** memoria del proyecto.
- **Fichero Diario.xlsx:** desglose de horas reales invertidas en el proyecto con el resumen total (presente en la memoria).
- **Fichero planificacion.mpp:** fichero de Microsoft Project con el diagrama de Gantt utilizado y explicado en la memoria del proyecto.

Cabe destacar que todo el sistema está puesto en **producción** en Heroku (servidor), GitHub Pages (página web) y Google Play (aplicación móvil). Ver Manual de Ejecución para más detalles.

11.2.2 Estructura del código fuente

11.2.2.1 Proyecto servidor OCAM

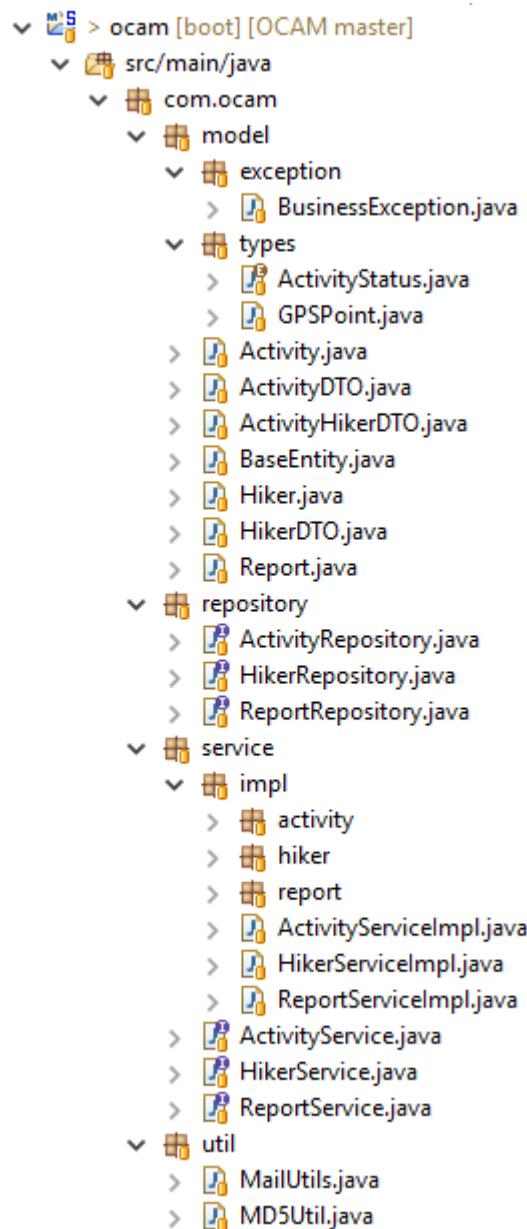
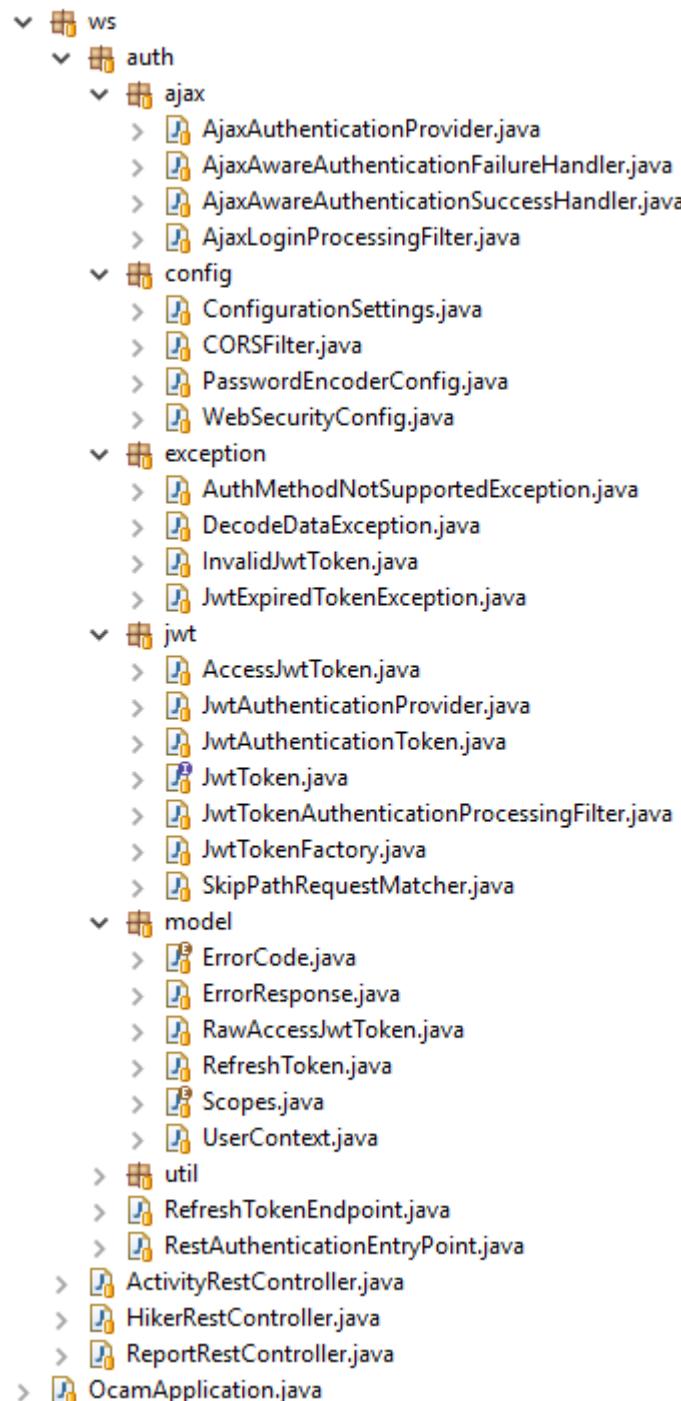


Figura 192 Estructura carpeta OCAM – src 1

- com.ocam.model: carpeta con las clases del dominio de la aplicación.
 - o com.ocam.model.exception: excepciones propias del dominio.
 - o com.ocam.model.types: clases enumeradas y tipos del dominio.
- com.ocam.repository: repositorios de la aplicación. Punto de acceso a la persistencia.
- com.ocam.service: servicios de negocio de la aplicación
- com.ocam.service.impl: implementación de los servicios de negocio: activity, hiker y report.
- com.ocam.util: Clases de utilidad para encriptar contraseñas utilizando el algoritmo de cifrado MD5 y enviar correos electrónicos por medio de java.mail.api.

*Figura 193 Estructura carpeta OCAM – src 2*

- com.ocam.ws.auth.ajax: filtro de entrada a la API para autenticarse en el sistema.
- com.ocam.ws.auth.config: clases de configuración programática de los filtros CORS, constantes para la generación de tokens, encriptación de passwords, filtros HTTP.
- com.ocam.ws.auth.exception: excepciones propias para el tratamiento de errores durante la autenticación.
- com.ocam.ws.auth.jwt: filtro de entrada a la API por medio de autenticación. Contienen las clases necesarias para la realización del filtro y la comprobación del token.

- com.ocam.ws.auth.model: clases de modelo, tipos y errores para el proceso de autenticación en el sistema.
- com.ocam.ws.auth.util: clases de utilidad y constantes del sistema.

11.2.2.2 Proyecto web OCAM-web

El proyecto OCAM-web está dividido en sus tres principales módulos: private, common y public.

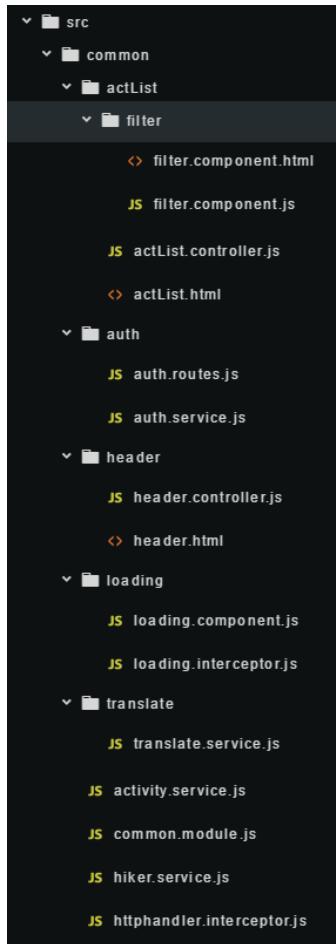


Figura 194 Estructura carpeta OCAM-web – src common

- common.actlist: vista del listado de actividades tanto pública como privada
- common.actlist.filter: componente de filtrado en la vista del listado de actividades
- common.auth: servicios de autenticación y comprobación de autenticación de rutas de la aplicación.
- common.header: vista y controlador del header de la aplicación
- common.loading: servicio de gestión e interceptor de eventos de carga de datos asíncronos de la aplicación
- common.translate: servicio que gestiona el i18n de la aplicación
- common: servicios de acceso a API para actividades y hikers. Interceptor de llamadas HTTP.

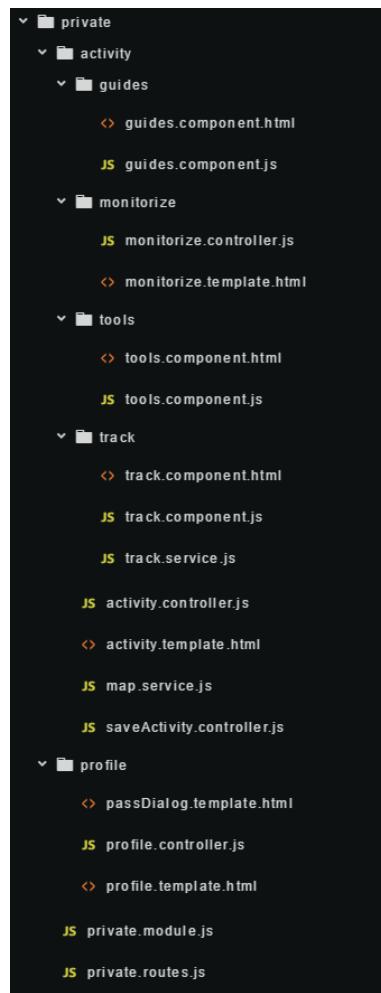


Figura 195 Estructura carpeta OCAM-web – src private

- private.activity: vistas parciales, componentes, servicios y controladores de la vista de una actividad.
- private.activity.guides: componente de asignación de guías a una actividad
- private.activity.monitorize: componente de monitorización de una actividad
- private.activity.tools: componente con accesos a gestión de una actividad
- private.activity.track: componente de visualización del track de una actividad
- private.profile: controlador y vista del perfil de un usuario

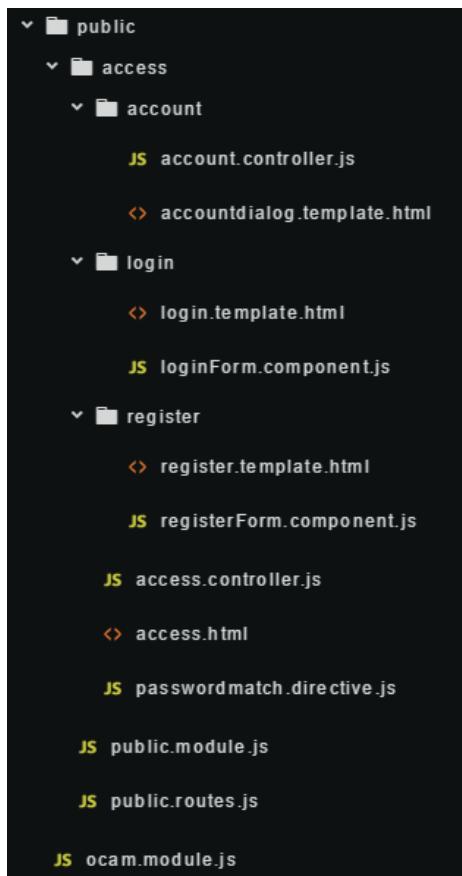


Figura 196 Estructura carpeta OCAM-web – src public

- public.access.account: controlador para activar una cuenta de usuario.
- public.access.login: controlador para acceder a la aplicación por medio del login.
- public.access.register: controlador para registrar un usuario nuevo en la aplicación.
- public: módulo público y gestión de rutas públicas.

11.2.2.3 Proyecto móvil OCAM-android

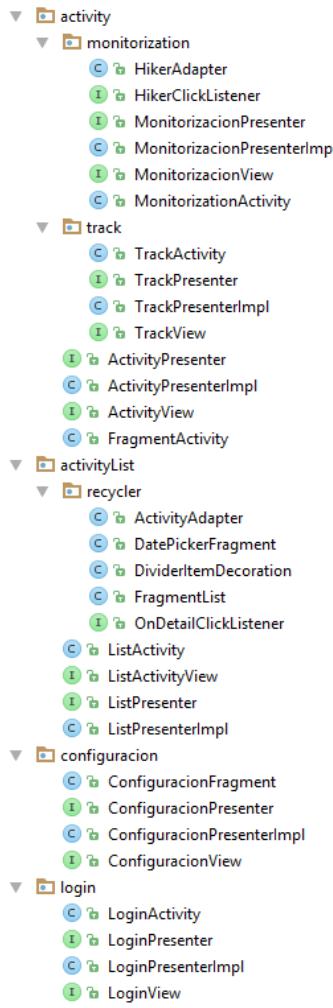


Figura 197 Estructura carpeta OCAM-android – src 1

- com.ocam.activity: actividad de Actividad del sistema
- com.ocam.activity.monitorization: actividad de monitorización de una actividad del sistema.
- com.ocam.activity.track: actividad de track de una actividad del sistema.
- com.ocam.activityList: actividad y fragment de la lista de actividades del sistema.
- com.ocam.activityList.recycler: recyclerView y adapter para el listado de actividades del sistema.
- com.ocam.configuracion: fragment para la configuración de la periodicidad de los reportes del usuario
- com.ocam.login actividad para hacer login en la aplicación.

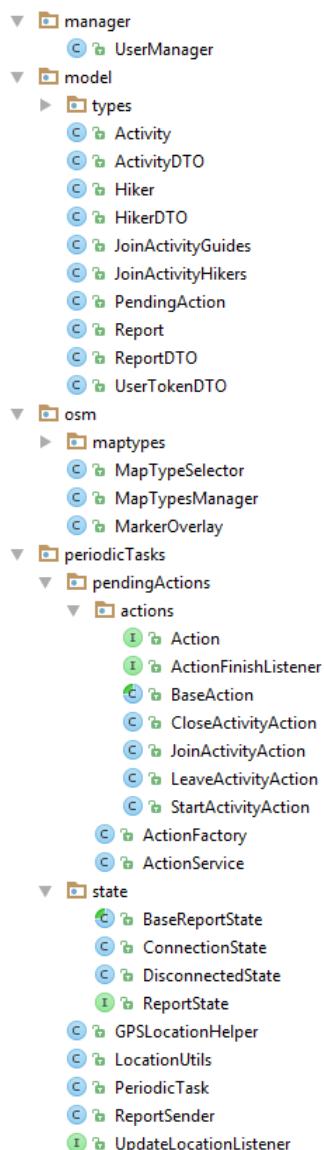


Figura 198 Estructura carpeta OCAM-android – src 2

- com.ocam.manager: manager de usuarios para almacenar el token y los datos del usuario identificado en el sistema
- com.ocam.model: clases del modelo anotadas del sistema y DTOs para transferencia de datos entre la vista y los presentadores.
- com.ocam.model.types: tipos y enumerados del modelo
- com.ocam.osm: clases que gestionan los proveedores de mapas
- com.ocam.osm.maptypes: clases por cada proveedor con métodos de donde extraer la información de las fotografías.
- com.ocam.periodicTasks: clases de ejecución en background para llevar a cabo la monitorización
- com.ocam.periodicTasks.pendingActions.actions: clases para gestionar las acciones pendientes de confirmación por falta de cobertura de datos.
- com.ocam.periodicTasks.state: comportamiento de envío de reportes según el estado en el que se encuentre el dispositivo

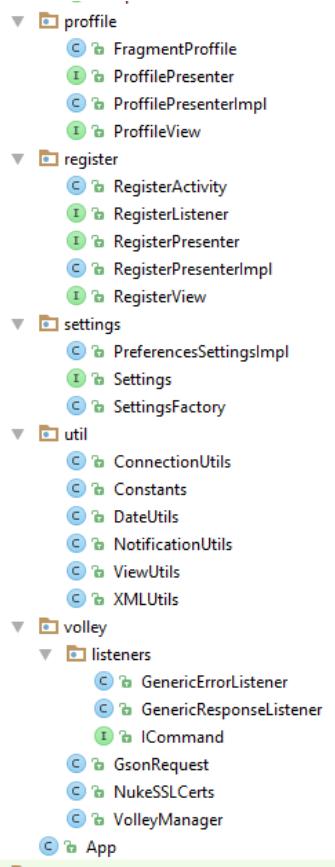


Figura 199 Estructura carpeta OCAM-android – src 3

- `com.ocam.profile`: actividad del perfil del usuario
- `com.ocam.register`: actividad para registrarse en el sistema
- `com.ocam.settings`: configuración del sistema con implementación en `SharedPreferences`.
- `com.ocam.util`: clases de utilidad
- `com.ocam.volley`: clases que gestionan peticiones HTTP
- `com.ocam.listeners`: clases que gestionan las respuestas a peticiones HTTP